SICS Research Report
R91:09

ISRN : SICS-R—91/09--SE
ISSN : 0283-3638

Strong normalizability in Martin-Löf´s Type Theory

by

Claes Löfwall and Gunnar Sjödin

May 1991

Swedish Institute of Computer Science
Box 1263, S-164 29 KISTA, SWEDEN

Strong normalizability in Martin-Löf's Type Theory

Clas Löfwall and Gunnar Sjödin

Swedish Institute of Computer Science
Box 1263, S-164 28, KISTA, Sweden

SICS research report R91:09

May 15, 1991

Abstract

In this paper we prove that any subexpression of a correct judgement in Martin-Löf's Type Theory is strongly normalizable. We use the well-established technique with a "computability predicate". The logic used in the proof is classical set theory.

Contents

1 Introduction

Constructive mathematics as it was developed by Bishop [Bis67] has got a formal basis in Per Martin-Löf's type theory. This theory contains four kinds of *judgements*, namely

- A is a category
- A and B are equal categories
- a is an element in A
- a and b are equal elements in A.

Martin-Löf has pointed out that all these judgements are decidable. In particular, it is decidable whether an expression is a proof of a proposition or not. This gives an algorithm for proof checking. The termination of the algorithm depends on the fact that an expression that occurs in a correct judgement has a value or, in other words, every such expression has a normal form. In this paper, we prove *strong* normalization, i.e., every reduction path from an expression in a correct judgement leads to a unique normal form. There is a lot of normalization proofs for different formal systems in the literature. In [Gir71], [ML71a], [ML71b] and [Pra71] Girard, Martin-Löf and Prawitz used a new technique for proving normalization, extending Tait's method, cf. [Tai67], by combining it with the Curry-Howard correspondence, i.e. the "formulas-as-types" principle, cf. [How80]. They defined a predicate (called "computable" by Martin-Löf) as a middle step in the proof. In [ML72] Martin-Löf used this technique to prove normalization for an earlier version of type theory (without the equality type and categories). Since a normalization proof is a kind of consistency proof, he has also claimed that normalization should follow from the semantic explanations of the judgements and the fact that all the inference rules are true with these explanations. But so far we have not seen such an argument which gives normalization for expressions with free variables.

Catarina Svensson has in [Sve90] given a normalization proof for a type theory including the equality type but not categories. Her notation is polymorphic, i.e., an expression does not contain information about its type. This is of no importance for the normalization proof, but to get the algorithm for proof checking mentioned above, the monomorphic notation is the relevant one. The proofs in [Sve90] and [ML72] differ in the definition of the computability predicate. It is easier in [Sve90] than in [ML72] to prove that the predicate is well-defined, but in neither of them there is an explicit proof of this.

In this paper we use induction over a transfinite "time" interval to define the computability predicate, and we prove that the predicate is well-defined. Our version of type theory includes categories and the equality type, but of the other standard types we only study the natural numbers, the function type and the

universe. We do not think that any essentially new phenomenon occurs when the other standard types are added.

In section 2 there is given a sequence of rules, which defines the notion of a *correct judgement* as something which is formally obtained by applying finitely many of the rules. The *reduction rules* are obtained from the rules named "*calc*", e.g., $app((x)b, a) \rightarrow b(x := a)$ and $T(n) \rightarrow N$. In the appendix we derive some properties, needed in the proof, of this formal system. However, for the so called "consistency property" we rely on [Coq90].

The purpose of this paper is to prove the following (cf. **theorem 7.1** on page 49):

> if *a is a category* or *a is an element in A* is a correct judgement, then a is strongly normalizable, i.e., any sequence of reductions from a will eventually stop.

We will rely on classical logic and set theory. For each correct judgement of the form "*A is a category*", the terms of type A is the set of expressions a such that $a \in A$ is correct. We define a subset $\Phi(A)$ of the set of terms of type A, which may be thought of as the set of "computable" terms of type A. We prove that if $a \in \Phi(A)$ then a is strongly normalizable, and that $\Phi(A)$ equals the set of terms of type A, if "*A is a category*" is correct.

We define the sets $\Phi(A)$ as the union of increasing sets $\varphi_t(A)$, where t belongs to a "time" interval of ordinal numbers (below the first uncountable ordinal). At each time t we apply one of the rules given in section 3 for which the premiss but not the conclusion is true at time t. At time 0, $\varphi_0(A) = \emptyset$ for all A. Among the conclusions are A *open* and A *finished*, which we have introduced to handle a problem in defining Φ. If $\varphi_t(A) \neq \emptyset$, then A *open* has been reached before time t. If A *finished* is reached at time t, it is true that $\varphi_s(A) = \varphi_t(A)$ for $s \geq t$ (we say that A is *stable*). This does not follow immediately from our rules, but is the content of one of our main lemmas. We need this property to be able to use an *induction principle* introduced in section 7, which depends on **lemma 6.3.1** on page 45 .

2 Correct Judgements

The following is a set of generating rules for correct judgements in an incomplete version of Per Martin-Löf's type theory. We include the formation of categories, but the only type formations we study are: The natural numbers N, the function-type $\Pi(A,B)$, the equality-type $Eq(A,a,b)$ and the type of all small types U. We write $A\ cat$, $A=B$, $a \in A$ and $a=b \in A$ for the four judgements

A is a category
A and B are equal categories
a is an element in A
a and b are equal elements in A

A judgement under assumptions is written $JUDGE\ (assump)$ where $JUDGE$ is one of the four judgement forms above and where $assump$ is a list of assumptions. An assumption list, $(x_1 \in A_1, \ldots, x_n \in A_n)$, may also be written in vector notation, $(\mathbf{x} \in \mathbf{A})$. There may be a hidden, constant, list of assumptions in all the judgements of a rule, except the rules with no premisses. This list is implicit to the left of all explicit assumptions. We will say that a judgement "is correct", if it is formally derivable from the rules given in 2.2, but usually we will omit the phrase "is correct". The category of all types is written set and the category of functions from A to B is written $(x \in A)B$. If $b \in B\ (x \in A)$ then the intended meaning of $(x)b$ is the corresponding function in $(x \in A)B$. If $b \in (x \in A)B$ and $a \in A$ then $app(b,a)$ is the element in B obtained by applying b to a. We have chosen the non-standard notation $app(b,a)$ (instead of $b(a)$) to be able to treat the formula $app((x)b,a)$ syntactically in the same way as the other maximum formulas. The successor function considered as an object of $(x \in N)N$ is written $(x)s(x)$, while s itself does not belong to the language. The natural number 1 is written $s(0)$ and it is a correct judgement that $s(0)=app((x)s(x),0) \in N$. The elimination forms corresponding to N, Π and Eq are written $N\text{-}elim$, $\Pi\text{-}elim$ and $Eq\text{-}elim$ respectively. The canonical element in $Eq(A,a,a)$ is written $id(A,a)$. The elements in U corresponding to N, Π, Eq are written n, π, eq respectively and the map from U to set is written T, so that e.g. $T(n) = N \in set$.

2.1 Syntactic equality and substitution

A *pointer expression*, p.e., is defined recursively by the following clauses

1. A variable is a p.e.

2. If f is a "constructor" (e.g. s, Eq, $N\text{-}elim$, app) of arity n, and $a_1, \ldots, a_n$ are p.e:s, then $f(a_1, \ldots, a_n)$ is a p.e.

3. If b is a p.e. and x is a variable, then we get a new p.e. obtained from $(*)b$ by adding pointers from $*$ to all occurrences of x in b and at the same time erasing the x:s. This p.e. is *denoted* by $(x)b$.

4. If a and b are p.e:s and x is a variable, then a new p.e. is obtained from $(* \in a)b$ by adding pointers from $*$ to all occurrences of x in a and b and at the same time erasing the x:s. This p.e. is *denoted* by $(x \in a)b$.

The pointer expressions are our syntactic units; by abuse of language we will, when convenient, refer to them simply as expressions. If b and a are (pointer) expressions then $b(x{:=}a)$ is the expression that is obtained by replacing every x in the pointer expression b with the pointer expression a. Using this definition, we see that if b' is a p.e., denoted by $(x)b$, and a is a p.e., then $b(x{:=}a)$ only depends on b' and a. Similarly, if b' is a p.e., denoted by $(x \in c)b$, and a is a p.e., then $b(x{:=}a)$ only depends on b' and a. Note that, if b is a p.e. and x and y are variables, where y does not occur in b, then $(x)b$ and $(y)b(x{:=}y)$ denote the same p.e..

2.2 Rules

2.2.1 Strong equality

=1

$$\frac{\mathrm{a} \in \mathrm{A}}{\mathrm{a} = \mathrm{a} \in \mathrm{A}}$$

=2

$$\frac{\mathrm{A\ cat}}{\mathrm{A} = \mathrm{A}}$$

=3

$$\frac{\mathrm{a} = \mathrm{b} \in \mathrm{A}}{\mathrm{b} = \mathrm{a} \in \mathrm{A}}$$

=4

$$\frac{\mathrm{A} = \mathrm{B}}{\mathrm{B} = \mathrm{A}}$$

=5

$$\frac{\mathrm{a} = \mathrm{b} \in \mathrm{A}\ ,\ \mathrm{b} = \mathrm{c} \in \mathrm{A}}{\mathrm{a} = \mathrm{c} \in \mathrm{A}}$$

=6

$$\frac{\mathrm{A} = \mathrm{B}\ ,\ \mathrm{B} = \mathrm{C}}{\mathrm{A} = \mathrm{C}}$$

=7

$$\frac{\mathrm{a} \in \mathrm{A}\ ,\ \mathrm{A} = \mathrm{B}}{\mathrm{a} \in \mathrm{B}}$$

=8

$$\frac{\mathrm{a} = \mathrm{b} \in \mathrm{A}\ ,\ \mathrm{A} = \mathrm{B}}{\mathrm{a} = \mathrm{b} \in \mathrm{B}}$$

2.2.2 General Rules

judge-assumption

$$\frac{\text{A cat}}{\text{x} \in \text{A (x} \in \text{A)}}$$

Here x is a "new" variable, i.e. x is not one of the variables in the hidden assumption list.

judge-assumption-addition

$$\frac{\text{A cat , JUDGE}}{\text{JUDGE (x} \in \text{A)}}$$

where x is a "new" variable.

2.2.3 judge-cat

judge-cat-1

$$\frac{}{\text{set cat}}$$

judge-cat-2

$$\frac{\text{A} \in \text{set}}{\text{A cat}}$$

judge-cat-2′

$$\frac{\text{A} = \text{B} \in \text{set}}{\text{A} = \text{B}}$$

judge-cat-3

$$\frac{\text{B cat (x} \in \text{A)}}{\text{(x} \in \text{A)B cat}}$$

judge-cat-3′

$$\frac{\text{A} = \text{A}' \text{ , B} = \text{B}' \text{ (x} \in \text{A)}}{\text{(x} \in \text{A)B} = \text{(x} \in \text{A}')\text{B}'}$$

judge-cat-4

$$\frac{\text{b} \in \text{B (x} \in \text{A)}}{\text{(x)b} \in \text{(x} \in \text{A)B}}$$

judge-cat-4′

$$\frac{\text{b} = \text{b}' \in \text{B (x} \in \text{A)}}{\text{(x)b} = \text{(x)b}' \in \text{(x} \in \text{A)B}}$$

judge-cat-5 (elimination-rule)

$$\frac{B\ \mathrm{cat}\ (x \in A)\ ,\ a \in A\ ,\ b \in (x \in A)B}{\mathrm{app}(b,a) \in B(x{:=}a)}$$

judge-cat-5′

$$\frac{a = a' \in A\ ,\ B\ \mathrm{cat}\ (x \in A)\ ,\ b = b' \in (x \in A)B}{\mathrm{app}(b,a) = \mathrm{app}(b',a') \in B(x{:=}a)}$$

judge-cat-6-calc

$$\frac{a \in A\ ,\ b \in B\ (x \in A)}{\mathrm{app}((x)b,a) = b(x{:=}a) \in B(x{:=}a)}$$

2.2.4 judge-N

judge-N-set

$$\frac{}{N \in \mathrm{set}}$$

judge-N-0

$$\frac{}{0 \in N}$$

judge-N-s

$$\frac{a \in N}{s(a) \in N}$$

judge-N-elim

$$\frac{\begin{array}{c}C \in (x \in N)\mathrm{set}\ ,\ a \in N\ ,\ b \in \mathrm{app}(C,0)\ ,\\ d \in (x \in N)(y \in \mathrm{app}(C,x))\mathrm{app}(C,s(x))\end{array}}{\text{N-elim}(C,a,b,d) \in \mathrm{app}(C,a)}$$

judge-N-0-calc

$$\frac{\begin{array}{c}C \in (x \in N)\mathrm{set}\ ,\ b \in \mathrm{app}(C,0)\ ,\\ d \in (x \in N)(y \in \mathrm{app}(C,x))\mathrm{app}(C,s(x))\end{array}}{\text{N-elim}(C,0,b,d) = b \in \mathrm{app}(C,0)}$$

judge-N-s-calc

$$\frac{\begin{array}{c}C \in (x \in N)\mathrm{set}\ ,\ a \in N,\ b \in \mathrm{app}(C,0)\ ,\\ d \in (x \in N)(y \in \mathrm{app}(C,x))\mathrm{app}(C,s(x))\end{array}}{\text{N-elim}(C,s(a),b,d) = \mathrm{app}(\mathrm{app}(d,a),\text{N-elim}(C,a,b,d)) \in \mathrm{app}(C,s(a))}$$

2.2.5 judge-Π

judge-Π-set

$$\frac{A \in \text{set} \ , \ B \in (x \in A)\text{set}}{\Pi(A,B) \in \text{set}}$$

judge-Π-λ

$$\frac{A \in \text{set} \ , \ B \in (x \in A)\text{set} \ , \ b \in (x \in A)\text{app}(B,x)}{\lambda(A,B,b) \in \Pi(A,B)}$$

judge-Π-elim

$$\frac{\begin{array}{l}A \in \text{set} \ , \ B \in (x \in A)\text{set} \ , \ C \in (x \in \Pi(A,B))\text{set} \ , \\ a \in \Pi(A,B), \ d \in (y \in (x \in A)\text{app}(B,x))\text{app}(C,\lambda(A,B,y))\end{array}}{\Pi\text{-elim}(A,B,C,a,d) \in \text{app}(C,a)}$$

judge-Π-λ-calc

$$\frac{\begin{array}{l}A \in \text{set} \ , \ B \in (x \in A)\text{set} \ , \ C \in (x \in \Pi(A,B))\text{set} \ , \\ b \in (x \in A)\text{app}(B,x) \ , \\ d \in (y \in (x \in A)\text{app}(B,x))\text{app}(C,\lambda(A,B,y))\end{array}}{\Pi\text{-elim}(A,B,C,\lambda(A,B,b),d) = \text{app}(d,b) \in \text{app}(C,\lambda(A,B,b))}$$

2.2.6 judge-Eq

judge-Eq-set

$$\frac{A \in \text{set} \ , \ a \in A \ , \ b \in A}{\text{Eq}(A,a,b) \in \text{set}}$$

judge-Eq-id

$$\frac{A \in \text{set} \ , \ a \in A}{\text{id}(A,a) \in \text{Eq}(A,a,a)}$$

judge-Eq-elim

$$\frac{\begin{array}{l}A \in \text{set} \ , \ a \in A \ , \ b \in A \ , \\ C \in (x \in A)(y \in A)(z \in \text{Eq}(A,x,y))\text{set} \ , \\ g \in \text{Eq}(A,a,b) \ , \ d \in (x \in A)\text{app}(\text{app}(\text{app}(C,x),x),\text{id}(A,x))\end{array}}{\text{Eq-elim}(A,a,b,C,g,d) \in \text{app}(\text{app}(\text{app}(C,a),b),g)}$$

judge-Eq-id-calc

$$\frac{\begin{array}{l}A \in \text{set} \ , \ C \in (x \in A)(y \in A)(z \in \text{Eq}(A,x,y))\text{set} \ , \\ e \in A \ , \ d \in (x \in A)\text{app}(\text{app}(\text{app}(C,x),x),\text{id}(A,x))\end{array}}{\begin{array}{c}\text{Eq-elim}(A,e,e,C,\text{id}(A,e),d) = \text{app}(d,e) \in \\ \in \text{app}(\text{app}(\text{app}(C,e),e),\text{id}(A,e))\end{array}}$$

2.2.7 judge-U

judge-U-set

$$\frac{}{\mathrm{U} \in \mathrm{set}}$$

judge-U-T

$$\frac{\mathrm{a} \in \mathrm{U}}{\mathrm{T(a)} \in \mathrm{set}}$$

judge-U-n

$$\frac{}{\mathrm{n} \in \mathrm{U}}$$

judge-U-n-calc

$$\frac{}{\mathrm{T(n)} = \mathrm{N} \in \mathrm{set}}$$

judge-U-π

$$\frac{\mathrm{a} \in \mathrm{U}\ ,\ \mathrm{b} \in (\mathrm{x} \in \mathrm{T(a)})\mathrm{U}}{\pi(\mathrm{a,b}) \in \mathrm{U}}$$

judge-U-π-calc

$$\frac{\mathrm{a} \in \mathrm{U}\ ,\ \mathrm{b} \in (\mathrm{x} \in \mathrm{T(a)})\mathrm{U}}{\mathrm{T}(\pi(\mathrm{a,b})) = \Pi(\mathrm{T(a)},(\mathrm{x})\mathrm{T(app(b,x))} \in \mathrm{set}}$$

judge-U-eq

$$\frac{\mathrm{a} \in \mathrm{U}\ ,\ \mathrm{b} \in \mathrm{T(a)}\ ,\ \mathrm{c} \in \mathrm{T(a)}}{\mathrm{eq(a,b,c)} \in \mathrm{U}}$$

judge-U-eq-calc

$$\frac{\mathrm{a} \in \mathrm{U}\ ,\ \mathrm{b} \in \mathrm{T(a)}\ ,\ \mathrm{c} \in \mathrm{T(a)}}{\mathrm{T(eq(a,b,c))} = \mathrm{Eq(T(a),b,c)} \in \mathrm{set}}$$

2.3 Definitions

Definition 2.3.1 (Reduction) *For any expressions* a *and* b

$$\mathrm{a} \longrightarrow \mathrm{b}$$

means that there is a subexpression c *of* a *and an instance of a rule above named "*-calc*" with conclusion of form* c = d ∈ A *and* b *is obtained by replacing* c *with* d *in* a.
Note that in the case of **judge-Π-λ-calc** *the left hand side of the rule is replaced by* Π-elim(A,B,C,λ(D,E,b),d) *and in the case of* **judge-Eq-id-calc** *the left hand side of the rule is replaced by* Eq-elim(A,a,b,C,id(D,e),d).
Furthermore,

$$a \xrightarrow{*} b$$

means that a $\longrightarrow \ldots \longrightarrow$ b *or* a = b.

■ (definition 2.3.1)

Definition 2.3.2 (Normality) *An expression* a *is said to be* normal *if there is no* b *such that* a $\longrightarrow$ b.

■ (definition 2.3.2)

Definition 2.3.3 (Strong normalizability) *An expression* a *is said to be* strongly normalizable, *abbreviated "* a **sn***", if every sequence*

$$a = a_0 \longrightarrow a_1 \longrightarrow a_2 \longrightarrow \ldots$$

is finite.

■ (definition 2.3.3)

Definition 2.3.4 (Convertibility) *The relation* conv *on expressions is the equivalence relation generated by "* $\longrightarrow$ *". As usual "expressions" should be read as "pointer expressions".*

■ (definition 2.3.4)

Lemma 2.3.1 (Church-Rosser) *Let* a,b,c *be expressions such that* a $\xrightarrow{*}$ b *and* a $\xrightarrow{*}$ c. *Then there is an expression* d *such that* b $\xrightarrow{*}$ d *and* c $\xrightarrow{*}$ d.

PROOF By the "Tait/Martin-Löf" method. See e.g. [ML72].

■ (lemma 2.3.1)

Definition 2.3.5 (Normal form) *Let a be an expression. If there is a reduction*

$$a \xrightarrow{*} b$$

such that b *is normal, then this* b *is denoted* $\overline{a}$. *Note that according to* **lemma 2.3.1** , $\overline{a}$ *is unique if it exists.*

■ (definition 2.3.5)

3 φ-rules and strong normalizability

The conclusions of the rules are of the type $a \in \varphi(A)$, A *open* or A *finished*, where a and A are expressions obtained from derivations in the formal system defined in section 2 extended by the following rules for constants, for $1 \leq n < \infty$

$$\frac{\text{A cat}}{A_n \in \text{A}}$$

Note that A is not considered to be a subexpression of A_n and hence the constants are normal. It follows that there are at most countably many possible conclusions, so it is enough to consider the time interval from 0 to the first uncountable ordinal. In the rules below we have omitted the index t in $\varphi_t(A)$. If a rule is applied at time t, $\varphi(A)$ above the line should be replaced by $\bigcup_{s<t} \varphi_s(A)$ and $\varphi(A)$ below the line by $\varphi_t(A)$.

If A *open* or A *finished* is proved at time t, it is considered to be true for all points of time after t.

An assumption a **elim** means that a is a constant or the leftmost form of the expression a is one of *app*, *T*, *N-elim*, Π-*elim*, *Eq-elim*. An assumption a **intro** means that a **elim** is false. An assumption a **eni** means that a is not a normal **intro**.

The following notations and conventions are used in the sequel:

F = The finished-rules, i.e. the rules which have a conclusion of the form A *finished*.

R = The rules below except the finished-rules and the φ-*completed* rule.

ALL-RULES = The rules below except the φ-*completed* rule.

We write e.g.

$$\text{A open} \prec \text{a} \in \varphi(\text{A})$$

to mean that A *open* is shown earlier than $a \in \varphi(A)$. Here A *open* and $a \in \varphi(A)$ are, by abuse of language, used for the ordinal t at which the respective properties are shown.

When we have a premiss P, written as a judgement, e.g. $a \in A$, this is to be understood as "P can be proved to be a correct judgement, using the rules for correct judgements". Note that we have no hidden assumptions in such judgements. Thus, the rules only produce statements about *finished*, *open* or φ for expressions without variables (recall that *expression* means *pointer expression* so a "free" variable is the same as a variable).

A rule *is applicable* at a certain time if the premisses of the rule are satisfied *and* the conclusion has not been shown. With this restriction on the use of the φ-rules, it follows that every proved conclusion is associated with exactly

one instance of a rule (though it may depend on the particular choices in our proof-path that we have made).

A rule is said to be *ready* at a point of time if its premisses are satisfied. Note that

$$\text{The rule X is applicable} \iff \text{The rule X is } \textit{ready} \text{ and its conclusion is not yet drawn}$$

A premiss of implication form such as

$$a \longrightarrow b \implies b \in \varphi(A)$$

in the rules below, should be read as: For every b such that $a \longrightarrow b$, we have $b \in \varphi(A)$.

3.1 The rules

Rules valid for every category:

general-Aφ

$$\frac{\begin{array}{l}A \text{ open }, \\ a \in A \text{ , a } \textbf{elim} \text{ , } (a \longrightarrow b) \implies b \in \varphi(A)\end{array}}{a \in \varphi(A)}$$

3.1.1 eni

Op-eni

$$\frac{A \text{ cat , A } \textbf{eni} \text{ , } (A \longrightarrow B) \implies B \text{ finished}}{A \text{ open}}$$

insert-eniφ (introduction rule for a category on eni form)

$$\frac{\begin{array}{l}A \text{ open}, \\ A \text{ cat , } A \longrightarrow B \text{ , } b \in \varphi(B) \text{ , b } \textbf{intro}\end{array}}{b \in \varphi(A)}$$

F-eni

$$\frac{\begin{array}{l}A \text{ open }, \\ A \text{ cat }, \\ A \textbf{ eni }, \\ \text{no rule in } \mathbf{R} \text{ is applicable}\end{array}}{A \text{ finished}}$$

3.1.2 cat

Eventually, we will interpret *A cat* with *A open* but we do not define φ*(cat)* (in case we would, it would consist of all categories *A* such that *A finished*). The following rules are about *A:s* such that *A cat.*

Op-cat A,B normal

$$\frac{\text{A finished , B cat } (x \in A) \text{ , } a \in \varphi(A) \implies B(x{:=}a) \text{ open}}{(x \in A)B \text{ open}}$$

(Note that to write *B(x:=a) finished* instead of *open* in the premisses would create problems e.g. when $B \equiv$ *set.*)

abstrφ A,B normal

$$\frac{\begin{array}{l}(x \in A)B \text{ open ,} \\ b \in B\ (x \in A) \text{ , } a \in \varphi(A) \implies b(x{:=}a) \in \varphi(B(x{:=}a))\end{array}}{(x)b \in \varphi((x \in A)B)}$$

F-cat A,B normal

$$\frac{\begin{array}{l}(x \in A)B \text{ open ,} \\ \text{no rule in } \mathbf{R} \text{ is applicable ,} \\ a \in \varphi(A) \implies B(x{:=}\ a) \text{ finished}\end{array}}{(x \in A)B \text{ finished}}$$

3.1.3 set

Op-set

$$\frac{}{\text{set open}}$$

set-insertφ

$$\frac{\begin{array}{l}\text{set open ,} \\ A \in \text{set , A finished , A } \mathbf{intro}\end{array}}{A \in \varphi(\text{set})}$$

F-set

$$\frac{\begin{array}{l}\text{set open ,}\\ \text{U finished ,}\\ \text{no rule in } \textbf{ALL-RULES} \setminus \{\text{F-U, F-set }\} \text{ is applicable}\end{array}}{\text{set finished}}$$

3.1.4 N

Op-N

$$\frac{}{\text{N open}}$$

0φ

$$\frac{\text{N open}}{0 \in \varphi(\text{N})}$$

sφ

$$\frac{\begin{array}{l}\text{N open ,}\\ \text{a} \in \varphi(\text{N})\end{array}}{\text{s(a)} \in \varphi(\text{N})}$$

F-N

$$\frac{\begin{array}{l}\text{N open ,}\\ \text{no rule in } \mathbf{R} \text{ is applicable}\end{array}}{\text{N finished}}$$

3.1.5 Π

Op-Π A,B normal

$$\frac{\text{A} \in \text{set , A finished , B} \in \varphi((\text{x} \in \text{A})\text{set}) \text{ , } (\text{x} \in \text{A})\text{app(B,x) finished}}{\Pi(\text{A,B}) \text{ open}}$$

$\lambda\varphi$ A,B normal

$$\frac{\begin{array}{l}A = A' \in \text{set}\ ,\ B = B' \in (x \in A)\text{set}\ ,\\ A'\ \mathbf{sn}\ ,\ B'\ \mathbf{sn}\ ,\\ \Pi(A,B)\ \text{open}\ ,\\ b \in \varphi((x \in A)\text{app}(B,x))\end{array}}{\lambda(A',B',b) \in \varphi(\Pi(A,B))}$$

F-Π A,B normal

$$\frac{\begin{array}{l}\Pi(A,B)\ \text{open}\\ \text{no rule in } \mathbf{R} \text{ is applicable}\end{array}}{\Pi(A,B)\ \text{finished}}$$

3.1.6 Eq

Op-Eq A,a,b normal

$$\frac{A \in \text{set}\ ,\ A\ \text{finished}\ ,\ a \in \varphi(A)\ ,\ b \in \varphi(A)}{\text{Eq}(A,a,b)\ \text{open}}$$

idφ A,a normal

$$\frac{\begin{array}{l}A = A' \in \text{set}\ ,\ a = a' \in A\ ,\\ A'\ \mathbf{sn}\ ,\ a' \in \varphi(A)\ ,\\ \text{Eq}(A,a,a)\ \text{open}\end{array}}{\text{id}(A',a') \in \varphi(\text{Eq}(A,a,a))}$$

F-Eq A,a,b normal

$$\frac{\begin{array}{l}\text{Eq}(A,a,b)\ \text{open}\\ \text{no rule in } \mathbf{R} \text{ is applicable}\end{array}}{\text{Eq}(A,a,b)\ \text{finished}}$$

3.1.7 U

Op-U

$$\frac{}{\text{U open}}$$

U-insertφ

$$\frac{\begin{array}{l}\text{U open ,}\\ \text{a} \in \text{U , T(a) finished , a } \textbf{intro}\end{array}}{\text{a } \epsilon\ \varphi(\text{U})}$$

F-U

$$\frac{\begin{array}{l}\text{U open}\\ \text{no rule in } \textbf{ALL-RULES} \setminus \{\text{F-U, F-set }\} \text{ is applicable}\end{array}}{\text{U finished}}$$

φ-completed

$$\frac{\text{no rule in } \textbf{ALL-RULES} \text{ is applicable}}{\varphi \textit{ completed}}$$

3.2 Strong normalizability

Definition 3.2.1 (Φ) *Let* t *be the time when* φ completed *is shown. For* A cat, *we let* $\Phi(\text{A}) = \varphi_t(\text{A})$ *and* A Φ-cat *mean that* A open *has been shown before time* t.

▓ (definition 3.2.1)

Lemma 3.2.1 (strong normalizability of computable terms) *If* d $\epsilon\ \Phi$(A) *or if* d Φ-cat *then* d *is strongly normalizable.*

PROOF It suffices to show, by induction on t, that

> If $d \epsilon \varphi(A)$ or if d *open* is shown at time t, then d is strongly normalizable.

We separate the proof into different cases, depending on how $d \epsilon \varphi(A)$ or d *open* is shown.

general-Aφ. For all b, we have

$$\text{d} \longrightarrow \text{b} \implies \text{b } \epsilon\ \varphi(\text{A}) \prec \text{d } \epsilon\ \varphi(\text{A})$$

Hence, by induction, all one-step reductions of d are strongly normalizable, which implies that d is strongly normalizable.

Op-eni. For all B, $d \longrightarrow B \Longrightarrow$ *B finished* $\prec$ *d open*. Thus, *B open* $\prec$ *B finished* $\prec$ *d open*, and hence, by induction, B is strongly normalizable. It follows that d is strongly normalizable.

insert-eniφ , d $\equiv$ b. Then $b \in \varphi(B) \prec b \in \varphi(A)$ and hence b is strongly normalizable.

Op-cat , d $\equiv$ (x $\in$ A)B. By assumption, A and B are strongly normalizable and hence so is d.

abstrφ , d $\equiv$ (x)b. Let $c \in A$ be a constant. Then, by **general-A**φ, $c \in \varphi(A)$ and therefore $b(x{:=}c) \in \varphi(A) \prec d \in \varphi((x \in A)B)$. By induction $b(x{:=}c)$ is strongly normalizable. Since c is a constant, it follows that b and hence d are strongly normalizable.

Op-set. Obvious.

set-insertφ, d $\equiv$ A. We have

$$\text{A open} \prec \text{A finished} \prec \text{A} \in \varphi(\text{set})$$

Hence, by induction A is strongly normalizable.

Op-N. Obvious.

0φ. Obvious.

sφ , d $\equiv$ s(a). By induction, a is strongly normalizable.

Op-Π , d $\equiv$ Π(A,B). By assumption, A and B are strongly normalizable and hence so is d.

$\lambda\varphi$, d $\equiv$ λ(A',B',b). By assumption A', B' are strongly normalizable and by induction b is strongly normalizable.

Op-Eq , d $\equiv$ Eq(A,a,b). By assumption, A, a and b are strongly normalizable and hence so is d.

Op-U. Obvious.

U-insertφ , d $\equiv$ a. We have

$$\text{T(a) open} \prec \text{T(a) finished} \prec \text{a} \in \varphi(\text{U}).$$

By induction, $T(a)$ is strongly normalizable and hence so is a.

∎ (lemma 3.2.1)

4 Stability of φ

Lemma 4.1 (φ(A) and Term(A)) *The following holds*

- *A open* $\Longrightarrow$ *A cat*
- $a \epsilon \varphi(A) \Longrightarrow a \in A$

PROOF We prove both statements by induction on time. The only rules that are non-trivial to check are handled below.

insert-elimφ
By induction, $b \in B$. According to **lemma A.18** on page 84 , $A = B$. Hence, by **=7**, $b \in A$.

Op-Π
By induction, $B \in (x \in A)set$. Hence, by **judge-Π-set** and **judge-cat-2**, $\Pi(A,B)$ *cat*.

$\lambda\varphi$
By induction, $b \in (x \in A)app(B,x)$ and $\Pi(A,B)$ *cat*. By **lemma A.15** on page 82 , $A \in set$ and $B \in (x \in A)set$. Hence, by **judge-Π-λ**, $\lambda(A,B,b) \in \Pi(A,B)$. Thus, by **lemma A.11** on page 79 and **lemma A.16** , $\lambda(A',B',b) \in \Pi(A,B)$.

idφ
By induction, $Eq(A,a,a)$ *cat* and $a' \in A$. By **lemma A.15** , $A \in set$, $a \in A$. Hence, by **judge-Eq-id**, **lemma A.11** and **lemma A.16** , $id(A',a') \in Eq(A,a,a)$.

∎ (lemma 4.1)

The above result will be used implicitly in the following.

Lemma 4.2 (Finishedness when A sn) *Let* A cat. *Suppose* A **sn**. *Then*

1. $\overline{A}$ *finished is shown* $\Longrightarrow$ *A finished is shown*
2. $\overline{A}$ *finished* $\prec$ *U finished* $\Longrightarrow$ *A finished* $\prec$ *U finished*

PROOF

1 If not, then there is a minimal element, D, in the reduction tree of A such that *D finished* is not shown. Since $D \not\equiv \overline{A}$, we must have D **eni**. But then, by **Op-eni**, *D open* is shown. It follows directly from **F-eni** that *D finished* is also shown.

2 Similarly, using the fact that **F-eni** is included in **ALL-RULES**.

▓ (lemma 4.2)

Lemma 4.3 (Finishedness in set) *Suppose that*

- $A \in set$
- A **sn**
- $\overline{A}$ *finished* $\prec$ U *finished*

Then

$A \epsilon \varphi(set) \prec U$ *finished*

PROOF Suppose that the statement is wrong. Then there is a minimal D in the reduction tree of A such that $D \epsilon \varphi(set) \prec U$ *finished* is not true. By **lemma A.19** on page 88 and **lemma 4.2**

D $\in$ set *and* D finished $\prec$ U finished

By **set-insert**φ, D **intro** is not true, i.e., D **elim**. Furthermore, by the minimality

D $\longrightarrow$ E $\Longrightarrow$ E ϵ φ(set) $\prec$ U finished

and hence, by **general-D**φ and **F-U**

D ϵ φ(set) $\prec$ U finished

Contradiction.

▓ (lemma 4.3)

Lemma 4.4 (Reductions in set) *Suppose that*

$A \epsilon \Phi(set)$, $A \xrightarrow{*} B$

Then

$B \epsilon \varphi(set) \preceq A \epsilon \varphi(set)$

PROOF It is enough to prove the statement when $A \longrightarrow B$. If A **elim** then the result follows from **general-set**φ. If A **intro** then B **intro** and, since A **eni**

B finished $\prec$ A open $\prec$ A finished

Hence, since **set-insert**φ is an **R**-rule

B ϵ φ(set) $\prec$ A finished

Also

A finished $\prec$ A ϵ φ(set)

and the result follows.

∎ (lemma 4.4)

Lemma 4.5 (Incorrect judgements) *There are no correct judgements of the following forms*

- $set \in D$
- $(x \in A)B \in D$

PROOF By an easy induction on the proof of a correct judgement.

∎ (lemma 4.5)

Definition 4.1 (Stability) *Suppose*

A finished is shown

and

$a \epsilon \Phi(A) \Longrightarrow a \epsilon \varphi(A) \prec A$ *finished*

i.e. $\varphi(A)$ *is not growing after A finished has been shown. Then A is said to be* stable.

∎ (definition 4.1)

Lemma 4.6 (Stabilizing lemma) *Suppose that*

A finished *is shown*

Then A *is* stable.

PROOF Suppose that the statement is false. Let G *finished* be minimal with respect to the property G is not *stable*. Let $d \epsilon \varphi(G)$ be minimal with respect to the property

$$d \in \varphi(G) \succ G \text{ finished}$$

We refer to the property of G as "the outer induction" and to the property of d as "the inner induction". Suppose that $d \in \varphi(G)$ has been shown with **general-G**φ. Thus, we have shown $d \in \varphi(G)$ with

$$\frac{G \text{ open} , d \in G , d \textbf{ elim} , (d \longrightarrow b) \Longrightarrow b \in \varphi(G)}{d \in \varphi(G)}$$

By the inner induction

$$(d \longrightarrow b) \Longrightarrow (b \in \varphi(G) \prec G \text{ finished})$$

It follows that the rule is *ready* at the time when we show *G finished.* Thus, its conclusion has been shown, which implies that $d \in \varphi(G) \prec G$ *finished*, i.e. a contradiction. Consequently, $d \in \varphi(G)$ is not shown with **general-G**φ. In particular, d **intro**.
We know that *G finished* has been shown with one of the following rules:

F-eni, F-cat, F-set, F-N, F-Π, F-Eq, F-U

In each case, we have to prove that the assumption "$d \in \varphi(G)$ is shown with an introduction rule for φ" gives a contradiction.

4.1 F-eni, G eni

The introduction rule is **insert-eni**φ

$$\frac{G \text{ open} , G \text{ cat} , G \longrightarrow B , d \in \varphi(B) , d \textbf{ intro}}{d \in \varphi(G)}$$

G open has been shown with **Op-eni** :

$$\frac{G \text{ cat} , (G \longrightarrow C) \Longrightarrow C \text{ finished}}{G \text{ open}}$$

Thus

$$B \text{ finished} \prec G \text{ open} \prec G \text{ finished}$$

By the outer induction B in **insert-eni**φ above, is *stable.* It follows that

$$d \in \varphi(B) \prec B \text{ finished} \prec G \text{ finished}$$

Hence, the rule is *ready* when we show *G finished.* It follows that its conclusion already has been drawn. Therefore,

$d \in \varphi(G) \prec G$ finished

which is a contradiction. Hence, *G finished* cannot have been shown by the use of **F-eni**.

* (F-**eni**)

4.2 F-cat, G $\equiv$ (x $\in$ A)B

The only φ-introduction rule is **abstr**φ

$$\frac{\begin{array}{l}(x \in A)B \text{ open} , \\ b \in B\ (x \in A) , a \in \varphi(A) \Longrightarrow b(x{:=}a) \in \varphi(B(x{:=}a))\end{array}}{(x)b \in \varphi((x \in A)B)}$$

where $d \equiv (x)b$.

Now, *A finished* $\prec$ *(x $\in$ A)B finished.* Thus, *A* is *stable* according to the outer induction. At the time when we are to show *(x $\in$ A)B finished*, we have

$a \in \Phi(A) \Longrightarrow$ (B(x:=a) finished $\prec$ (x $\in$ A)B finished)

It follows that *B(x:=a)* is *stable* for $a \in \Phi(A)$. Thus, the premisses for $d \in \varphi(G)$ are satisfied at this time and hence the conclusion has been drawn. We infer that *G finished* cannot have been shown with **F-cat**.

* (F-cat)

4.3 F-set, G $\equiv$ set

The only φ-introduction rule is **set-insert**φ

$$\frac{\begin{array}{l}\text{set open} , \\ d \in \text{set} , d \text{ finished} , d \textbf{ intro}\end{array}}{d \in \varphi(\text{set})}$$

It follows that

$d \in \varphi(\text{set}) \succ d$ finished $\succ$ set finished

Take the minimal in the set of

e finished

such that

e $\in$ set *and* e finished $\succ$ set finished

We refer to the property of *e* as "induction on *e*". We have several cases depending on which of the following rules *e finished* has been shown with:

F-eni , F-N, F-Π, F-Eq eller F-U.

The rules **F-set** and **F-cat** are not candidates because of **lemma 4.5** .

4.3.1 F-eni

By induction and **lemma A.19** on page 88

$$e \longrightarrow b \implies b \text{ finished} \prec e \text{ finished} \implies$$
$$\implies b \text{ finished} \prec \text{set finished}$$

Thus, *e open* $\prec$ *set finished*, and hence by **F-eni** (and **F-set**)

$$e \text{ finished} \prec \text{set finished}$$

which is a contradiction.

4.3.2 F-N, e ≡ N

Op-N has no premisses. It follows that *N open* $\prec$ *set finished* and hence by **F-N**

$$N \text{ finished} \prec \text{set finished}$$

Contradiction.

4.3.3 F-Π, e ≡ Π(A,B) A,B normal

In order to get a contradiction, it suffices, by the premisses of **F-Π** and **F-set**, to prove that

$$\Pi(A,B) \text{ open} \prec \text{set finished}$$

To do this, it is enough to prove that

1. A finished $\prec$ set finished

2. $B \in \varphi((x \in A)\text{set}) \prec$ set finished

3. $(x \in A)\text{app}(B,x)$ finished $\prec$ set finished

We know that

a. $A \in$ set , A finished $\prec e$ finished

b. $B \in \varphi((x \in A)\text{set}) \prec e$ finished

c. $(x \in A)\text{app}(B,x)$ finished $\prec e$ finished

The proof of the statements **1-3** follows.

1. From **a** above and induction on e.

2. By **1**

$$(x \in A)\text{set open} \prec \text{set finished}$$

If B **elim** then the result follows by **general-(x ∈ A)set**φ since B is normal. Otherwise

$$B \in \varphi((x \in A)\text{set})$$

has been shown with

$$\frac{\begin{array}{l}(x \in A)\text{set open} ,\\ b \in \text{set } (x \in A) , a \in \varphi(A) \Longrightarrow b(x{:=}a) \in \varphi(\text{set})\end{array}}{(x)b \in \varphi((x \in A)\text{set})}$$

where $B \equiv (x)b$. We know by **1** that *A finished* $\prec$ *set finished* and hence, by the outer induction, A is *stable*. Let $a \in \Phi(A)$. Then

$$b(x{:=}a) \in \varphi(\text{set}) \prec B \in \varphi((x \in A)\text{set}) \prec e \text{ finished} \preceq$$
$$\preceq d \text{ finished} \prec d \in \varphi(\text{set})$$

so that by the inner induction

$$b(x{:=}a) \in \varphi(\text{set}) \prec \text{set finished}$$

It follows that the rule is *ready* at the time when we are to show *set finished* and hence the conclusion is already drawn i.e. **2** is proved.

3. If B is not of the form $(x)B'$ then $(x \in A)app(B,x)$ is normal and, by **Op-cat** and **F-cat**, it suffices to prove (remember that A is *stable*)

$$a \in \Phi(A) \Longrightarrow \text{app(B,a) finished} \prec \text{set finished}$$

Let $a \in \Phi(A)$. We have

$$\text{app(B,a) finished} \prec (x \in A)\text{app(B,x) finished} \prec e \text{ finished}$$

Thus, by induction on e

$$\text{app(B,a) finished} \prec \text{set finished}$$

Note that, by **judge-cat-5** and **lemma 4.1** on page 20 , $app(B,a) \in set$. In the other case $B \equiv (x)B'$. There is only one reduction

$$(x \in A)\text{app(B,x)} \equiv (x \in A)\text{app((x)B',x)} \longrightarrow (x \in A)B'$$

It follows from **Op-eni** and **F-eni** that it is sufficient to prove that

$$(x \in A)B' \text{ finished} \prec \text{set finished}$$

Since $B \in (x \in A)set$ it follows from **lemma A.15** on page 82 that

$B' \in \text{set}\ (x \in A)$

Thus, it suffices to prove that

$a \in \Phi(A) \implies B'(x:=a)\ \text{finished} \prec \text{set finished}$

Let $a \in \Phi(A)$. Then by **lemma A.6** on page 66 and **lemma 4.1** on page 20 , $B'(x:=a) \in set$. Furthermore

$B'(x:=a)\ \text{finished} \prec (x \in A)B'\ \text{finished} \prec$
$(x \in A)\text{app}(B,x)\ \text{finished} \prec e\ \text{finished}$

Thus, by the minimality of e

$B'(x:=a)\ \text{finished} \prec \text{set finished}$

We have obtained a contradiction of the assumption that *e finished* has been shown with F-Π.

4.3.4 F-Eq, e ≡ Eq(A,a,b)

It suffices to show that *Eq(A,a,b) open* $\prec$ *set finished.* By induction on e

$A\ \text{finished} \prec \text{set finished}$

Thus, by the outer induction A is *stable* and hence

$a \in \varphi(A) \prec A\ \text{finished}$
$b \in \varphi(A) \prec A\ \text{finished}$

It follows that

$a \in \varphi(A) \prec \text{set finished}$
$b \in \varphi(A) \prec \text{set finished}$

and hence

$\text{Eq}(A,a,b)\ \text{open} \prec \text{set finished}$

We are done.

4.3.5 F-U, e ≡ U

F-set has in its premisses *U finished*, which directly gives a contradiction.

We have now reached a contradiction of the assumption that *G finished* is shown with **F-set**.

* (F-set)

4.4 F-N, G $\equiv$ N

The φ-introduction rules are $\mathbf{0}\varphi$ and $\mathbf{s}\varphi$.

$\mathbf{0}\varphi$, d $\equiv$ 0 : Apparently, this is impossible.

$\mathbf{s}\varphi$, d $\equiv$ s(a) : By the inner induction, $a \in \varphi(N) \prec N$ *finished*. In the usual way we conclude that $s(a) \in \varphi(N) \prec N$ *finished*. We have obtained a contradiction of the assumption that *G finished* is shown with **F-N**.

* (F-N)

4.5 F-Π, G $\equiv$ Π(A,B)

The only φ-introduction rule is $\lambda\varphi$

$$\frac{\begin{array}{l}A = A' \in \text{set} \ , \ B = B' \in (x \in A)\text{set} \ , \\ A' \ \mathbf{sn} \ , \ B' \ \mathbf{sn} \ , \\ \Pi(A,B) \text{ open} \ , \\ b \in \varphi((x \in A)\text{app}(B,x))\end{array}}{\lambda(A',B',b) \in \varphi(\Pi(A,B))}$$

where $d \equiv \lambda(A',B',b)$.

We have

(x $\in$ A)app(B,x) finished $\prec$ Π(A,B) open $\prec$
$\prec$ Π(A,B) finished = G finished

By the outer induction $(x \in A)app(B,x)$ is *stable* and hence

b $\in$ φ((x $\in$ A)app(B,x)) $\prec$ (x $\in$ A)app(B,x) finished $\prec$
$\prec$ G finished

It follows that the rule above for $d \in \varphi(G)$ is *ready* at the time, when we show *G finished*. It follows that the conclusion has already been drawn, which gives

d $\in$ φ(G) $\prec$ G finished

We have obtained a contradiction of the assumption that *G finished* is shown with **F-Π**.

* (F-Π)

4.6 F-Eq, G $\equiv$ Eq(A,a,b)

In the case when $a \not\equiv b$ then there is no introduction rule. If $a \equiv b$ then the only φ-introduction rule is $\mathbf{id}\varphi$

$$\frac{\begin{array}{l}A = A' \in \text{set} \ , \ a = a' \in A \ , \\ A' \ \mathbf{sn} \ , \ a' \in \varphi(A) \ , \\ \text{Eq}(A,a,a) \text{ open}\end{array}}{\text{id}(A',a') \in \varphi(\text{Eq}(A,a,a))}$$

where $d \equiv id(A',a')$.

Now,

A finished $\prec$ Eq(A,a,a) finished

and hence by the outer induction, we obtain that A is *stable*. It follows that

a' ϵ φ(A) $\prec$ A finished

Thus, the conclusion in the above rule must be shown before *Eq(A,a,a) finished*. We get a contradiction of the assumption that *G finished* has been shown with **F-Eq**.

* (F-Eq)

4.7 F-U, G ≡ U

The only φ-introduction rule is **U-insert**φ

$$\frac{\begin{array}{l}\text{U open ,}\\ \text{d} \in \text{U , T(d) finished , d } \textbf{intro}\end{array}}{\text{d } \epsilon\ \varphi(\text{U})}$$

Obviously, *T(d) finished* $\succ$ *U finished* . By **lemma 3.2.1** on page 18 , *T(d)* is strongly normalizable, so that according to **lemma 4.2** on page 20

$\overline{T(d)}$ finished $\succ$ U finished

Take the minimal in the set of

$\overline{T(e)}$ finished

such that $e \in U$ and

$\overline{T(e)}$ finished $\succ$ U finished

Note that this choice of e does not imply that $\overline{e}$ exists. Neither is e **intro** necessarily true. To get a contradiction, we have to show that in fact

$\overline{T(e)}$ finished $\prec$ U finished

By **Op-eni**, **F-eni** and **F-U** , we have $\overline{T(e)}$ **intro**. Hence, during the reduction $T(e) \xrightarrow{*} \overline{T(e)}$ the "T" must have been resolved. This means that the reduction may be written

T(e) $\xrightarrow{*}$ T(f) $\xrightarrow{*}$ $\overline{T(e)}$, where e $\xrightarrow{*}$ f

and where f has an introduction form "belonging to" T. Note that, according to **lemma A.19** on page 88 , $f \in U$. Thus, we may assume that e is on one of the forms

n , π**(a,b)** , **eq(a,b,c)**

4.7.1 e ≡ n

$\overline{T(n)} \equiv N$ and N *finished* $\prec U$ *finished.*

4.7.2 e ≡ π(a,b)

$$\overline{T(e)} \equiv \overline{T(\pi(a,b))} \equiv \Pi(\overline{T(a)},(x)\overline{T(app(b,x))})$$

Note that we have used **lemma 2.3.1** on page 12 to obtain e.g. that $\overline{T(a)}$ exists. Let $B \equiv T(app(b,x))$. Here

$$\Pi(\overline{T(a)},(x)\overline{B}) \text{ open}$$

must have been shown with **Op-Π**. We know that

a. $\overline{T(a)}$ finished $\prec \overline{T(e)}$ finished

b. $(x)\overline{B} \in \varphi((x \in \overline{T(a)})\text{set}) \prec \overline{T(e)}$ finished

c. $(x \in \overline{T(a)})\text{app}((x)\overline{B},x)$ finished $\prec \overline{T(e)}$ finished

We have to prove that

1. $\overline{T(a)}$ finished $\prec$ U finished

2. $(x)\overline{B} \in \varphi((x \in \overline{T(a)})\text{set}) \prec$ U finished

3. $(x \in \overline{T(a)})\text{app}((x)\overline{B},x)$ finished $\prec$ U finished

1. The result follows from **a** and the the minimality of $\overline{T(e)}$. Note that $a \in U$ by **lemma A.15** on page 82 . In particular, by the outer induction, $\overline{T(a)}$ is *stable.*

2. By **1**

$$(x \in \overline{T(a)})\text{set open} \prec \text{U finished}$$

Now

$$\overline{B} \in \text{set } (x \in \overline{T(a)})$$

Here we have used **lemma A.12** on page 80 , **lemma A.15** , **lemma A.18** and **lemma A.19** . Thus, by **abstr**φ, it is sufficient to prove that

$$c \in \Phi(\overline{T(a)}) \Longrightarrow \overline{B}(x:=c) \in \varphi(\text{set}) \prec \text{U finished}$$

Let $c \in \Phi(\overline{T(a)})$. We know that

$$\overline{B}(x:=c) \in \varphi(\text{set}) \prec (x)\overline{B} \in \varphi((x \in \overline{T(a)})\text{set}) \prec$$
$$\prec \overline{T(e)} \text{ finished}$$

and hence in particular, by **lemma 3.2.1** , that $\overline{B}(x:=c)$ is *strongly normalizable* . Hence, by **lemma 4.3** on page 21 , it is sufficient to prove that

$\overline{\overline{B}(x := c)}$ finished $\prec$ U finished

But this is true if $\overline{\overline{B}(x := c)}$ **eni**. Thus we may assume that it is a normal **intro**. Now

T(app(b,c)) $\equiv$ B(x:=c) $\xrightarrow{*}$ $\overline{B}$(x:=c)

so that

$\overline{\overline{B}(x := c)} \equiv \overline{T(app(b, c))}$

But $app(b,c) \in U$ and hence, by the minimality of $\overline{T(e)}$ *finished* , it suffices to prove that

$\overline{\overline{B}(x := c)}$ finished $\prec \overline{T(e)}$ finished

Now, using **lemma 4.4** on page 21

$\overline{\overline{B}(x := c)}$ finished $\prec \overline{\overline{B}(x := c)} \in \varphi$(set) $\preceq$
$\preceq \overline{B}$(x:=c) $\in \varphi$(set) $\prec \overline{T(e)}$ finished

and we are done.

3. Since the only reduction of $(x \in \overline{T(a)})app((x)\overline{B},x)$ is

(x $\in \overline{T(a)}$)app((x)$\overline{B}$,x) $\longrightarrow$ (x $\in \overline{T(a)}$)$\overline{B}$

it suffices to prove that

c $\in \Phi(\overline{T(a)}) \Longrightarrow \overline{B}$(x:=c) finished $\prec$ U finished

Let $c \in \Phi(\overline{T(a)})$. We proved in **2** that $\overline{B}(x:=c)$ is strongly normalizable. Thus, according to **lemma 4.2** on page 20 , it is sufficient to prove that

$\overline{\overline{B}(x := c))}$ finished $\prec$ U finished

and this was also done in **2.**

4.7.3 e ≡ eq(a,b,c)

$\overline{T(e)} \equiv$ Eq($\overline{T(a)}$,$\overline{b}$,$\overline{c}$)

Here $Eq(\overline{T(a)},\overline{b},\overline{c})$ *open* must have been shown with **Op-Eq**. It is sufficient to show that

Eq($\overline{T(a)}$,$\overline{b}$,$\overline{c}$) open $\prec$ U finished

We know that

$\overline{T(a)}$ finished $\prec \overline{T(e)}$ finished
$\overline{b} \in \varphi(\overline{T(a)}) \prec \overline{T(e)}$ finished
$\overline{c} \in \varphi(\overline{T(a)}) \prec \overline{T(e)}$ finished

By the minimality of $\overline{T(e)}$

$\overline{T(a)}$ finished $\prec$ U finished

Hence, by the outer induction $\overline{T(a)}$ is *stable* so that

$\overline{b} \in \varphi(\overline{T(a)}) \prec \overline{T(a)}$ finished
$\overline{c} \in \varphi(\overline{T(a)}) \prec \overline{T(a)}$ finished

whence the result follows.

This gives a total contradiction of the assumption about "e" and hence a total contradiction of the assumption that G *finished* has been shown with **F-U**.

* (F-U)

We have gone through all the possibilities for G. Thus, we have obtained a contradiction of the assumption that the lemma is not true.

∎ (lemma 4.6)

5 Finishedness and reduction of φ

Lemma 5.1 (finished-lemma) *When φ is* completed, *then*

$$A \text{ open} \implies A \text{ finished}$$

i.e. if A open *is shown then* A finished *is shown too.*

PROOF Let *C open* be minimal with respect to the property "*C finished* is never shown". We have to consider the different **Op**-rules, which may have shown *C open*. By the form of the **F**-rules, it is obvious that *C finished* is shown unless the **Op**-rule is **Op-cat** or **Op-set**. By **Op-U** and **F-U**, *U finished* will be shown. Thus, by **F-set**, the **Op**-rule cannot be **Op-set**. It remains to consider the case

Op-cat , C $\equiv$ (x $\in$ A)B

We have to show that the corresponding instance of **F-cat** will be used. This is obviously the case if we eventually get

$$a \in \Phi(A) \implies B(x:=a) \text{ finished}$$

However, since *A finished* $\prec$ *C open* , **lemma 4.6** on page 22 gives that *A* is *stable*. Hence

$$a \in \Phi(A) \implies B(x:=a) \text{ open} \prec C \text{ open}$$

By induction, this means that we eventually will show that *B(x:=a) finished* for every $a \in \Phi(A)$.

∎ (lemma 5.1)

Lemma 5.2 (Reduction lemma) *We have the following reduction properties*

1. A *Φ-cat* , $A \longrightarrow B \implies B$ *Φ-cat*
2. $d \in \Phi(C)$, $d \longrightarrow e \implies e \in \Phi(C)$

PROOF
The statement in **1** is obvious since $A \longrightarrow B$ implies that A **eni** and hence *A open* must have been shown with **Op-eni**.
Now consider **2**. Suppose that the statement is false. Then there is a minimal

$$d \in \varphi(C)$$

such that d does not have this property. We separate the proof into different cases, depending on how $d \in \varphi(C)$ has been shown.

general-Aφ, d $\equiv$ a , C $\equiv$ A
Let $a \longrightarrow e$. According to the premisses in the rule

$$e \in \varphi(A) \prec a \in \varphi(A)$$

so that in particular

$$e \in \Phi(A)$$

Contradiction.

insert-elimφ, d $\equiv$ b , C $\equiv$ A
We have

$$b \in \varphi(B) \prec d \in \varphi(C)$$

Let $b \longrightarrow e$. Then, by induction $e \in \Phi(B)$. Thus **insert-elimφ** is *ready* at the time when we use the *φ-completed* rule. It follows that

$$e \in \Phi(A)$$

abstrφ, d $\equiv$ (x)b , C $\equiv$ (x $\in$ A)B
Reduction by

$$(x)b \longrightarrow (x)b' \text{ , } \textit{where } b \longrightarrow b'$$

By **lemma 4.6** on page 22

$$a \in \Phi(A) \Longrightarrow b(x{:=}a) \in \varphi(B(x{:=}a)) \prec d \in \varphi(C)$$

so that, by induction

$$a \in \Phi(A) \Longrightarrow b'(x{:=}a) \in \Phi(B(x{:=}a))$$

Note that according to **lemma A.19** on page 88 , $b' \in B\ (x \in A)$. Thus, **abstr-φ** will eventually be used to show

$$(x)b' \in \varphi((x \in A)B)$$

set-insertφ, d $\equiv$ A , C $\equiv$ set
$A \longrightarrow A'$. Thus A **eni** and A *open* must have been shown with **Op-eni** which implies that A' *finished* is shown. Since A **intro**, so is A', and it follows that

$$A' \in \varphi(set)$$

is shown.

0φ , d $\equiv$ 0 , C $\equiv$ N
Obvious.

$s\varphi$, $\mathbf{d} \equiv \mathbf{s(a)}$, $\mathbf{C} \equiv \mathbf{N}$
Reduction by

$$s(a) \longrightarrow s(a'), \textit{ where } a \longrightarrow a'$$

We have

$$a \in \varphi(N) \prec s(a) \in \varphi(N)$$

By induction $a' \in \Phi(N)$ and hence, by $s\varphi$ used on a', $s(a') \in \Phi(N)$.

$\lambda\varphi$, $\mathbf{d} \equiv \lambda\mathbf{(A',B',b)}$, $\mathbf{C} \equiv \Pi\mathbf{(A,B)}$
The reductions are

1. $\lambda(A',B',b) \longrightarrow \lambda(A'',B',b)$, where $A' \longrightarrow A''$
2. $\lambda(A',B',b) \longrightarrow \lambda(A',B'',b)$, where $B' \longrightarrow B''$
3. $\lambda(A',B',b) \longrightarrow \lambda(A',B',b')$, where $b \longrightarrow b'$

We prove below that the reductions, in the separate cases, lie in $\Phi(\Pi(A,B))$

1. By **lemma A.18** on page 84 , $A = A'' \in set$. Furthermore, since A' **sn**, so is A''. It follows that we eventually will show
   $$\lambda(A'',B',b) \in \varphi(\Pi(A,B))$$
   with $\lambda\varphi$.
2. Analogous to **1**.
3. By induction
   $$b' \in \Phi((x \in A)app(B,x))$$
   and hence
   $$\lambda(A',B',b') \in \varphi(\Pi(A,B))$$
   will be shown.

We have reached a contradiction in the case $\lambda\varphi$.

$\mathbf{id}\varphi$, $\mathbf{d} \equiv \mathbf{id(A',a')}$, $\mathbf{C} \equiv \mathbf{Eq(A,a,a)}$
The reductions are

1. $id(A',a') \longrightarrow id(A'',a')$, *where* $A' \longrightarrow A''$
2. $id(A',a') \longrightarrow id(A',a'')$, *where* $a' \longrightarrow a''$

Since, $A = A'' \in set$ and A'' **sn**, case **1** follows directly. In the case of **2**, we have $a = a'' \in A$. Furthermore

$$a' \in \varphi(A) \prec d \in \varphi(C)$$

so that by induction, $a'' \in \Phi(A)$. Thus, we will eventually show

id(A′,a″) ϵ φ(Eq(A,a,a))

with **id**φ.

U-insertφ, **d** $\equiv$ **a** , **C** $\equiv$ **U**

We have a reduction $a \longrightarrow a'$. Now

T(a′) finished $\prec$ T(a) finished , a′ **intro**

and hence we will eventually show $a' \in \varphi(U)$ with **U-insert**φ.

▓ (lemma 5.2)

6 Properties of Φ

6.1 Auxiliary results concerning Φ

Lemma 6.1.1 (Invariance under reduction) *Suppose*

A Φ-*cat* and $A \xrightarrow{*} B$

Then

B Φ-*cat* and $\Phi(A) = \Phi(B)$

PROOF By **lemma 3.2.1** on page 18 , A is strongly normalizable. Furthermore, by **lemma 5.2** on page 33 , D Φ-*cat* for every D in the reduction tree of A. Thus, it suffices to prove that every element, D, in the reduction tree of A satisfies

$$\Phi(\mathrm{D}) = \Phi(\overline{A})$$

Suppose not. Then there is a minimal element, D, in the reduction tree not satisfying this property. By minimality

$$\mathrm{D} \longrightarrow \mathrm{E} \implies \Phi(\mathrm{E}) = \Phi(\overline{A})$$

If the statement is false then there is either an element in $\Phi(D)$ not in $\Phi(\overline{A})$ or vice versa. In the first case there is a minimal

$$\mathrm{a} \in \varphi(\mathrm{D})$$

such that

$$\mathrm{a} \notin \Phi(\overline{A})$$

By the premisses in **insert-eni**φ, $a \in \varphi(D)$ cannot have been shown with this rule. Thus it is shown with **general-D**φ. By minimality

$$\mathrm{a} \longrightarrow \mathrm{b} \implies \mathrm{b} \in \Phi(\overline{A})$$

and hence, using **general-**$\overline{A}\varphi$, we conclude that $a \in \Phi(\overline{A})$, which is a contradiction of the first case.
In the second case there is a minimal

$$\mathrm{a} \in \varphi(\overline{A})$$

such that

$$\mathrm{a} \notin \Phi(\mathrm{D})$$

If a **intro** then, by **insert-eni**φ used on a reduction $D \longrightarrow E$, $a \in \varphi(D)$ must be shown. Hence, a **elim** and $a \in \varphi(\overline{A})$ must have been shown with **general-**$\overline{A}\varphi$. By minimality

$$a \longrightarrow b \implies b \in \Phi(D)$$

and hence, using **general-D**φ, we conclude that $a \in \Phi(D)$, which is a contradiction of the second case.

■ (lemma 6.1.1)

Lemma 6.1.2 (Invariance under equality) *Suppose that*

- *A* Φ*-cat, B* Φ*-cat*
- $A = B$

Then

$$\Phi(A) = \Phi(B)$$

PROOF By **lemma A.20** on page 89 , *A conv B* . Hence, by **lemma 2.3.1** on page 12 , $\overline{A} \equiv \overline{B}$ so that by **lemma 6.1.1**

$$\Phi(A) = \Phi(\overline{A}) = \Phi(B)$$

■ (lemma 6.1.2)

Lemma 6.1.3 (Reductions and Φ-cat) *Let* A cat *and let*

$$A \xrightarrow{*} B$$

Then the following statements are equivalent

1. *A* **sn**, *B* Φ*-cat*

2 *A* Φ*-cat*

PROOF

1 $\implies$ **2** By **lemma 6.1.1** on page 37 , $\overline{B} \equiv \overline{A}$ is Φ-cat and hence, by **lemma 5.1** on page 33 and **lemma 4.2** on page 20 *A* Φ-cat.

2 $\implies$ **1** This follows from **lemma 3.2.1** on page 18 and **lemma 6.1.1** on page 37 .

■ (lemma 6.1.3)

Lemma 6.1.4 (Φ-cat and Φ(set)) *Let* $A \in$ set. *Then*

$A\ \Phi\text{-}cat \iff A \in \Phi(set)$

PROOF

$\Longrightarrow$

Suppose that A *open* is minimal with respect to the property

$$A \notin \Phi(\text{set})$$

By **lemma 5.1** on page 33 A *finished* is shown. By **set-insert**φ, A **intro** cannot be true. Thus, A **elim** and A *open* has been shown with **Op-eni**. Suppose $A \longrightarrow B$. Then

$$\text{B open} \prec \text{B finished} \prec \text{A open}$$

Thus, by induction

$$\text{B} \in \Phi(\text{set})$$

Now, **general-set**φ gives a contradiction.

$\Longleftarrow$

Suppose $A \in \varphi(set)$ is minimal with respect to the property that A *open* is not shown.

Suppose that A **intro**. Then **insert-eni**φ cannot have been used to show $A \in \varphi(set)$, since *set* is normal. Hence, **set-insert**φ must have been used to show it. Thus A *finished* is shown. Contradiction. Suppose now that A **elim** and $A \longrightarrow B$. Then

$$\text{B} \in \varphi(\text{set}) \prec \text{A} \in \varphi(\text{set})$$

By induction, B Φ-*cat* so that by **lemma 5.1** on page 33 B *finished* is shown. Hence, eventually, **Op-eni** may be applied to show A *open*. Contradiction.

▓ (lemma 6.1.4)

Lemma 6.1.5 (Membership to cat) $a \in \Phi(A) \Longrightarrow A\ \Phi\text{-}cat$

PROOF If $a \in \varphi(A)$ has been shown then A *open* has been shown, since it is a part of the premisses of all relevant rules.

▓ (lemma 6.1.5)

6.2 Φ-rules

For every rule in **R** (defined in section 3) we obtain a Φ-rule by taking away the assumptions of normality and making the following replacements

- $\varphi \longrightarrow \Phi$
- *open* $\longrightarrow$ Φ-*cat*
- *finished* $\longrightarrow$ Φ-*cat*

The corresponding rule is renamed by replacing **Op** by **OP** and φ by Φ. In particular, this gives us the following Φ-rules:

general-AΦ

$$\frac{\begin{array}{l}\mathrm{A}\ \Phi\text{-cat}\ , \\ \mathrm{a} \in \mathrm{A}\ ,\ \mathrm{a}\ \mathbf{elim}\ ,\ (\mathrm{a} \longrightarrow \mathrm{b}) \implies \mathrm{b} \in \Phi(\mathrm{A})\end{array}}{\mathrm{a} \in \Phi(\mathrm{A})}$$

OP-cat

$$\frac{\mathrm{A}\ \Phi\text{-cat}\ ,\ \mathrm{B}\ \mathrm{cat}\ (\mathrm{x} \in \mathrm{A})\ ,\ \mathrm{a} \in \Phi(\mathrm{A}) \implies \mathrm{B}(\mathrm{x}{:=}\mathrm{a})\ \Phi\text{-cat}}{(\mathrm{x} \in \mathrm{A})\mathrm{B}\ \Phi\text{-cat}}$$

abstrΦ

$$\frac{\begin{array}{l}(\mathrm{x} \in \mathrm{A})\mathrm{B}\ \Phi\text{-cat}\ , \\ \mathrm{b} \in \mathrm{B}\ (\mathrm{x} \in \mathrm{A})\ ,\ \mathrm{a} \in \Phi(\mathrm{A}) \implies \mathrm{b}(\mathrm{x}{:=}\mathrm{a}) \in \Phi(\mathrm{B}(\mathrm{x}{:=}\mathrm{a}))\end{array}}{(\mathrm{x})\mathrm{b} \in \Phi((\mathrm{x} \in \mathrm{A})\mathrm{B})}$$

OP-N

$$\frac{}{\mathrm{N}\ \Phi\text{-cat}}$$

0Φ

$$\frac{\mathrm{N}\ \Phi\text{-cat}}{0 \in \Phi(\mathrm{N})}$$

sΦ

$$\frac{\begin{array}{l}\mathrm{N}\ \Phi\text{-cat}\ , \\ \mathrm{a} \in \Phi(\mathrm{N})\end{array}}{\mathrm{s}(\mathrm{a}) \in \Phi(\mathrm{N})}$$

OP-Π

$$\frac{A \in \text{set} \ , \ A\ \Phi\text{-cat} \ , \ B \in \Phi((x \in A)\text{set}) \ , \ (x \in A)\text{app}(B,x)\ \Phi\text{-cat}}{\Pi(A,B)\ \Phi\text{-cat}}$$

λΦ

$$\frac{\begin{array}{l} A = A' \in \text{set} \ , \ B = B' \in (x \in A)\text{set} \ , \\ A' \ \mathbf{sn} \ , \ B' \ \mathbf{sn} \ , \\ \Pi(A,B)\ \Phi\text{-cat} \ , \\ b \in \Phi((x \in A)\text{app}(B,x)) \end{array}}{\lambda(A',B',b) \in \Phi(\Pi(A,B))}$$

OP-Eq

$$\frac{A \in \text{set} \ , \ A\ \Phi\text{-cat} \ , \ a \in \Phi(A) \ , \ b \in \Phi(A)}{\text{Eq}(A,a,b)\ \Phi\text{-cat}}$$

idΦ

$$\frac{\begin{array}{l} A = A' \in \text{set} \ , \ a = a' \in A \ , \\ A' \ \mathbf{sn} \ , \ a' \in \Phi(A) \ , \\ \text{Eq}(A,a,a)\ \Phi\text{-cat} \end{array}}{\text{id}(A',a') \in \Phi(\text{Eq}(A,a,a))}$$

OP-U

$$\frac{}{U\ \Phi\text{-cat}}$$

U-insertΦ

$$\frac{\begin{array}{l} U\ \Phi\text{-cat} \ , \\ a \in U \ , \ T(a)\ \Phi\text{-cat} \ , \ a\ \mathbf{intro} \end{array}}{a \in \Phi(U)}$$

Lemma 6.2.1 (Correctness of Φ-rules) *All the Φ-rules above are valid, i.e. if the premisses above the line in a Φ-rule are satisfied then the conclusion below the line is also satisfied.*

PROOF We demonstrate the proof method in some examples.

OP-cat

Suppose that the premisses in this Φ-rule are satisfied. By **lemma 3.2.1** on page 18 , A and $B(x{:=}a)$ are strongly normalizable for all $a \epsilon \Phi(A)$. By choosing a as a constant, we see that B and hence also $(x \in A)B$ are strongly normalizable. Thus, by **lemma 6.1.3** on page 38 , it suffices to prove that

$$(\mathrm{x} \in \overline{A})\overline{B}\ \Phi\text{-cat}$$

By **lemma 6.1.1** on page 37

$$\mathrm{a} \epsilon \Phi(\overline{A}) \Longrightarrow \mathrm{a} \epsilon \Phi(\mathrm{A}) \Longrightarrow \mathrm{B(x{:=}a)}\ \Phi\text{-cat} \Longrightarrow \Longrightarrow \overline{B}(\mathrm{x{:=}a})\ \Phi\text{-cat}$$

By **lemma 5.1** on page 33 , **lemma A.12** on page 80 , **lemma A.18** and **lemma A.19** , the following is true at the time for proving *φ-completed*

$$\overline{A}\ \text{finished}\ ,\ \overline{B}\ \text{cat}\ (\mathrm{x} \in \overline{A})\ ,\ \mathrm{a} \epsilon \varphi(\overline{A}) \Longrightarrow \overline{B}(\mathrm{x{:=}a})\ \text{open}$$

i.e. the premisses in **Op-cat**, needed to draw the conclusion $(x \in \overline{A})\overline{B}$ *open*, are satisfied. Thus, the conclusion is already shown. It follows that

$$(\mathrm{x} \in \mathrm{A})\mathrm{B}\ \Phi\text{-cat}$$

abstrΦ

Suppose that the premisses in this Φ-rule are satisfied. By **lemma 3.2.1** on page 18 , A and B are strongly normalizable. Hence by **lemma 6.1.1** on page 37 , it suffices to prove that

$$(\mathrm{x})\mathrm{b} \epsilon \Phi((\mathrm{x} \in \overline{A})\overline{B})$$

By the time when *φ-completed* is shown we have

$$\mathrm{a} \epsilon \varphi(\overline{A}) \Longrightarrow \mathrm{a} \epsilon \Phi(\overline{A}) \Longrightarrow \mathrm{a} \epsilon \Phi(\mathrm{A}) \Longrightarrow \Longrightarrow \mathrm{b(x{:=}a)} \epsilon \Phi(\mathrm{B(x{:=}a)}) \Longrightarrow \mathrm{b(x{:=}a)} \epsilon \Phi(\overline{B}(\mathrm{x{:=}a})) \Longrightarrow \Longrightarrow \mathrm{b(x{:=}a)} \epsilon \varphi(\overline{B}(\mathrm{x{:=}a}))$$

Furthermore, as above $b \in \overline{B}\ (x \in \overline{A})$. Hence the premisses in **abstrφ** are *ready* so the conclusion

$$(\mathrm{x})\mathrm{b} \epsilon \varphi((\mathrm{x} \in \overline{A})\overline{B})$$

has already been drawn.

OP-Π

Suppose that the premisses in this Φ-rule are satisfied. By **lemma 3.2.1** on page 18 , A and B are strongly normalizable. By **lemma 5.2** on page 33 , **lemma 6.1.1** on page 37 and **lemma A.19** on page 88

$\overline{A} \in$ set , $\overline{A}$ Φ-cat , $\overline{B} \in \Phi((x \in \overline{A})\text{set})$, $(x \in \overline{A})\text{app}(\overline{B},x)$ Φ-cat

Thus, when *φ-completed* is shown we have

$\overline{A} \in$ set , $\overline{A}$ finished , $\overline{B} \in \varphi((x \in \overline{A})\text{set})$,
$(x \in \overline{A})\text{app}(\overline{B},x)$ finished

so the conclusion $\Pi(\overline{A},\overline{B})$ *open* has already been shown, i.e. $\Pi(\overline{A},\overline{B})$ *Φ-cat* . It follows from **lemma 6.1.3** on page 38 that $\Pi(A,B)$ *Φ-cat* .

$\lambda\Phi$

By **lemma 3.2.1** on page 18 , A and B are strongly normalizable. By **lemma 5.2** on page 33 and **lemma 6.1.1** on page 37

$\Pi(\overline{A},\overline{B})$ Φ-cat , $b \in \Phi((x \in \overline{A})\text{app}(\overline{B},x))$

Thus, when *φ-completed* is shown we have

$\overline{A} = A' \in$ set , $\overline{B} = B' \in (x \in \overline{A})$set
A' **sn** , B' **sn**
$\Pi(\overline{A},\overline{B})$ open
$b \in \varphi((x \in \overline{A})\text{app}(\overline{B},x))$

so the conclusion

$\lambda(A',B',b) \in \varphi(\Pi(\overline{A},\overline{B}))$

has already been shown, i.e.

$\lambda(A',B',b) \in \Phi(\Pi(\overline{A},\overline{B}))$

By **lemma 6.1.1** on page 37

$\lambda(A',B',b) \in \Phi(\Pi(A,B))$

▓ (lemma 6.2.1)

6.3 Induction methods

For A *cat* correct, let *Term(A)* denote the set of expressions a such that $a \in A$ is correct.

Definition 6.3.1 *Let* D cat, *where* D $\equiv$ $(x \in A)B$ *or* D *has one of the forms*

N , **Π** , **Eq** , **U**

Consider the subsets I *of* Term(D). *We will say that I is closed under rules of type*

$$\frac{\textit{condition}}{\textit{expression} \in I}$$

if "expression" is in I, *whenever "condition" is satisfied. For every D as above , we will associate a number of such rules called generating-rules for D. For every D*

general-DI:

$$\frac{a \in D\ ,\ a\ \textbf{elim}\ ,\ (a \longrightarrow b) \Longrightarrow b \in I}{a \in I}$$

is one of the generating-rules. Furthermore, for each form we associate additional generating-rules, called introduction rules, as follows:

- **cat** - *for subsets I of Term((x ∈ A)B) , (x ∈ A)B Φ-cat , A,B normal*

abstrI:

$$\frac{b \in B\ (x \in A)\ ,\ a \in \Phi(A) \Longrightarrow b(x{:=}a) \in \Phi(B(x{:=}a))}{(x)b \in I}$$

- **N** - *for subsets I of Term(N)*

0I:

$$\frac{}{0 \in I}$$

sI:

$$\frac{a \in I}{s(a) \in I}$$

- **Π** - *for subsets I of Term(Π(A,B)) , Π(A,B) Φ-cat , A,B normal*

λI:

$$\frac{\begin{array}{l} A = A' \in \textit{set}\ ,\ B = B' \in (x \in A)\textit{set}\ , \\ A'\ \textbf{sn}\ ,\ B'\ \textbf{sn}\ , \\ b \in \Phi((x \in A)\textit{app}(B,x)) \end{array}}{\lambda(A',B',b) \in I}$$

- **Eq** - *for subsets I of Term(Eq(A,a,b)) , Eq(A,a,b) Φ-cat , A,a,b normal*

idI: - *in the case when* $a \equiv b$

$$\frac{A = A' \in set\ ,\ a = a' \in A \quad A'\ \mathbf{sn}\ ,\ a' \epsilon \Phi(A)}{id(A',a') \epsilon I}$$

- **U** - *for subsets I of Term(U)*

U-insertI:

$$\frac{a \in U\ ,\ a\ \mathbf{intro}\ ,\ T(a)\ \Phi\text{-}cat}{a \epsilon I}$$

▓ (definition 6.3.1)

Lemma 6.3.1 (Induction lemma) *Let* D cat *and suppose that* D open *is shown with one of the open-rules named* cat, N, Π, Eq, U. *Then* Φ(D) *is the least subset* I *of Term(D), which is closed under the generating-rules for* D.

PROOF We consider different cases depending on the form of *D*. By **lemma 4.1** on page 20 , $\Phi(D)$ is a subset of *Term(D)*. That, in each case, Φ(D) respects the rules follows directly from **lemma 6.2.1** on page 41 . Suppose $J \subseteq Term(D)$ is closed under the generating-rules for *D*. We want to show that $\Phi(D) \subseteq J$. Let $d \epsilon \Phi(D)$. We show that $d \epsilon J$ by induction on the time when $d \epsilon \varphi(D)$ is proved. If $d \epsilon \varphi(D)$ has been shown with **general-D**φ, then

$$(d \longrightarrow b) \Longrightarrow b \epsilon \varphi(D)$$

But then, by induction, $b \epsilon J$ and hence by **general-DI** $d \epsilon J$. Thus, in the following we do not have to repeat this argument in each separate case.

D $\equiv$ **(x** $\in$ **A)B**

Here $d \epsilon \varphi((x \in A)B)$ has been shown with **abstr**φ. Hence, $d \equiv (x)b$ and when we apply the rule the following is true

- $b \in B(x \in A)$ *and* $a \epsilon \varphi(A) \Longrightarrow b(x:=a) \epsilon \varphi((x \in A)B)$
- A finished

By **lemma 4.6** on page 22 , it follows that

$$a \epsilon \Phi(A) \Longrightarrow b(x:=a) \epsilon \Phi((x \in A)B)$$

We conclude that $(x)b \epsilon J$.

D $\equiv$ **N**

1. $d \in \varphi(N)$ is shown with 0φ. But $0 \in J$.

2. $d \in \varphi(N)$ is shown with $s\varphi$. Thus, $a \in \varphi(N) \prec s(a) \in \varphi(N)$. By induction, $a \in J$. It follows that $s(a) \in J$.

D ≡ Π(A,B)
Here $d \equiv \lambda(A',B',b) \in \varphi(\Pi(A,B))$ is shown with $\lambda\varphi$. Hence

$$b \in \Phi((x \in A)app(B,x))$$

so that $\lambda(A',B',b) \in J$.

D ≡ Eq(A,a,b)
In case $a \not\equiv b$, there is no introduction rule. Otherwise

$$d \equiv id(A',a') \in \varphi(Eq(A,a,a))$$

is shown with $\mathbf{id}\varphi$. Therefore $id(A',a') \in J$.

D ≡ U
Here $d \in \varphi(U)$ is shown with **U-insert**φ and hence *T(d) finished*. It follows that *T(d) Φ-cat* and we conclude that $d \in J$.

▓ (lemma 6.3.1)

Lemma 6.3.2 (Elim criterion in Φ) *Let* A Φ-cat *and let*

$$F \equiv E(b_1,\ldots,a,\ldots,b_n) \in A$$

be an elimination form with a in the main branch. Furthermore, suppose that $b_1,\ldots,b_n$ *all are strongly normalizable and that either*

a. *The main branch a is itself an elimination form and*

$$(a \longrightarrow a') \Longrightarrow E(b_1,\ldots,a',\ldots,b_n) \in \Phi(A)$$

or

b. *The main branch* $a \equiv \text{intr}(c_1,\ldots,c_m)$ *, where* intr *is an introduction form "belonging to"* E*, and where* $c_1,\ldots,c_m$ *are all strongly normalizable and the resolution of the main redex in* F *is in* Φ(A).

Then $F \in \Phi(A)$.

PROOF

a. Suppose that F is not in Φ(A). Consider the reduction tree of F. We have to show that every one-step reduction is in Φ(A). According to the assumptions, such a reduction is in Φ(A) if the reduction is performed in *"a"*. Thus, there must be a reduction in F, in one of the side branches, which is not in Φ(A). Consider the reduced reduction tree of F, which is generated by only reducing in the side branches. By assumption, this tree is finite. It follows that there is a minimal element in it, which is not in Φ(A). Suppose that we have reached this element by:

$$\mathrm{E}(b_1,\ldots,a,\ldots,b_n) \xrightarrow{*} \mathrm{E}(b_1',\ldots,a,\ldots,b_n')$$

Because of the minimality of $E(b_1',\ldots,a,\ldots,b_n')$, all its reductions in the side branches are in Φ(A). Now, consider the main branch reduction

$$\mathrm{E}(b_1',\ldots,a,\ldots,b_n') \longrightarrow \mathrm{E}(b_1',\ldots,a',\ldots,b_n')$$

We have

$$\mathrm{E}(b_1,\ldots,a,\ldots,b_n) \longrightarrow \mathrm{E}(b_1,\ldots,a',\ldots,b_n) \xrightarrow{*} \mathrm{E}(b_1',\ldots,a',\ldots,b_n')$$

and hence, by **lemma 5.2** on page 33

$$\mathrm{E}(b_1',\ldots,a',\ldots,b_n') \in \Phi(\mathrm{A})$$

which shows that all one-step reductions of $E(b_1',\ldots,a,\ldots,b_n')$ are in Φ(A). We have reached a contradiction.

b. Suppose that F is not in Φ(A). Consider the reduction tree of F. We have to show that every one-step reduction is in Φ(A). According to the assumptions, such a reduction is in Φ(A) if the reduction is the reduction of *F:s* main redex. Thus, there must be a reduction in F, which is not the reduction of the main redex, and which is not in Φ(A). Consider the reduced reduction tree of F, which is generated by only reducing redexes which are not the main redex. By assumption, this tree is finite. It follows that there is a minimal element in it, which is not in Φ(A). Suppose that we have reached this element by:

$$\mathrm{E}(b_1,\ldots,\mathrm{intr}(c_1,\ldots,c_m),\ldots,b_n) \xrightarrow{*} \mathrm{E}(b_1',\ldots,\mathrm{intr}(c_1',\ldots,c_m'),\ldots,b_n')$$

Because of the minimality of $E(b_1',\ldots,intr(c_1',\ldots,c_m'),\ldots,b_n')$ all its reductions, avoiding the main redex, are in Φ(A). Let G be the result of the (schematic) resolution of the main redex of $\mathrm{E}(x_1,\ldots,\mathrm{intr}(y_1,\ldots,y_m),\ldots,x_n)$ Now, consider the main redex reduction

$$\mathrm{E}(b_1',\ldots,\mathrm{intr}(c_1',\ldots,c_m'),\ldots,b_n') \longrightarrow$$
$$\longrightarrow \mathrm{G}(x_1{:=}b_1',\ldots,\ y_1{:=}c_1',\ldots,\ y_m{:=}c_m',\ldots,\ x_n{:=}b_n')$$

We have

$$\begin{aligned}&\mathrm{E}(b_1,\ldots,\mathrm{intr}(c_1,\ldots,c_m),\ldots,b_n) \longrightarrow \\ &\longrightarrow \mathrm{G}(x_1{:=}b_1,\ldots,\ y_1{:=}c_1,\ldots,\ y_m{:=}c_m,\ldots,\ x_n{:=}b_n) \xrightarrow{*} \\ &\xrightarrow{*}\ \mathrm{G}(x_1{:=}b_1',\ldots,\ y_1{:=}c_1',\ldots,\ y_m{:=}c_m',\ldots,\ x_n{:=}b_n')\end{aligned}$$

and hence by **lemma 5.2** on page 33

$$\mathrm{G}(x_1{:=}b_1',\ldots,\ y_1{:=}c_1',\ldots,\ y_m{:=}c_m',\ldots,\ x_n{:=}b_n') \in \Phi(\mathrm{A})$$

which proves that all one-step reductions of $E(b_1',\ldots,intr(c_1',\ldots,c_m'),\ldots,b_n')$ are in $\Phi(\mathrm{A})$. We have reached a contradiction.

■ (lemma 6.3.2)

7 The main theorem

Theorem 7.1 *If we have a correct judgement of one of the following forms*

- A *cat*
- $A = B$
- $a \in A$
- $a = b \in A$

then, for each case, the occurring expressions are strongly normalizable.

7.1 Preliminaries

By **lemma A.11** on page 79 , it suffices to prove that if a *cat* or $a \in A$ then a is strongly normalizable. In order to prove this, we need to introduce the concept of ultra-correct judgements.

Definition 7.1.1 (ultra-correct) *A judgement*

$$JUDGE \ \ (x_1 \in A_1, \ldots, x_n \in A_n)$$

is said to be ultra-correct if it is correct and satisfies the following "extra requirements", inductively defined by induction on the length n *of the assumption list,*

n = 0

Then we have a judgement without assumptions. The extra requirement indicated by "$\rightsquigarrow$ " is given for the respective different cases of the form of the judgement in the following way:

- A *cat* $\rightsquigarrow$ A Φ-*cat.*
- $a \in A \ \rightsquigarrow \ a \in \Phi(A)$.
- $A = B \ \rightsquigarrow$ A *cat* is ultra-correct, B *cat* is ultra-correct.
- $a = b \in A \ \rightsquigarrow$ $a \in A$ is ultra-correct, $b \in A$ is ultra-correct.

n > 0

The extra requirements for the judgement

$$JUDGE \ \ (x_1 \in A_1, \ldots, x_n \in A_n)$$

is that

- A_1 *is ultra-correct*

- *if* $a_1 \epsilon \Phi(A_1)$, *then*

  $JUDGE(x_1{:=}a_1)\ (x_2 \in A_2(x_1{:=}a_1), \ldots, x_n \in A_n(x_1{:=}a_1))$

  is ultra-correct.

∎ (definition 7.1.1)

PROOF of the theorem Assume that $a \in A\ \ (x_1 \in A_1, \ldots, x_n \in A_n)$ is a correct judgement. By **lemma 7.1** below, the judgement is ultra-correct. By repeated use of the inductive definition of *ultra-correctness*, we may choose constants $a_1, \ldots, a_n$ such that the following is true

A_1 cat and Φ-cat, $a_1 \in \Phi(A_1)$
$A_2(x_1{:=}a_1)$ cat and Φ-cat, $a_2 \epsilon \Phi(A_2(x_1{:=}a_1))$
...
$A_n(x_1{:=}a_1)\ldots(x_{n-1}{:=}a_{n-1})$ cat and Φ-cat, $a_n \epsilon \Phi(A_n(x_1{:=}a_1)\ldots(x_{n-1}{:=}a_{n-1}))$

$a(x_1{:=}a_1)\ldots(x_n{:=}a_n) \epsilon \Phi(\mathrm{A}((x_1{:=}a_1)\ldots(x_n{:=}a_n))$

By **lemma 3.2.1** on page 18 , $a(x_1{:=}a_1)\ldots(x_n{:=}a_n)$ is strongly normalizable and hence a is strongly normalizable. The proof that a is strongly normalizable if the form of the judgement is *a cat* is analogous.

∎ (theorem 7.1)

It remains to prove the lemma referred to in the proof.

Lemma 7.1 (Ultra-correctness lemma) *Every correct judgement is ultra-correct.*

PROOF See the next section.

∎ (lemma 7.1)

7.2 Proof of the ultra-correctness lemma

We first make the following

Definition 7.2.1 *A rule is said to preserve ultra-correctness if*

the premisses are ultra-correct $\Longrightarrow$
$\Longrightarrow$ *the conclusion is ultra-correct.*

∎ (definition 7.2.1)

To prove **lemma 7.1** , it will suffice to prove that every rule preserves ultra-correctness. We prove this by induction on the length of the hidden assumption list in an instance of a rule. The inductive step is easily established, by using the inductive definition of ultra-correctness. Note that this can be done for each rule individually. Thus, we only need to prove the base case, i.e. when there is no hidden assumption list. Before we start, we first explain the inner induction principle, which is to be used on the elimination rules. We then proceed to prove the base case for each individual rule.

7.2.1 Principles for the inner induction

Let *D cat* have one of the forms

(x ∈ A)B , N , Π(A,B) , Eq(A,a,b) , U

with the respective elimination forms

app , N-elim , Π-elim , Eq-elim , T

For such a D with elimination form $E(b_1,\ldots,a,\ldots,b_n)$, where a is the main branch and where we will know that $b_1,\ldots,b_n$ all are "in Φ", and hence by **lemma 3.2.1** strongly normalizable, we let

$$\mathrm{I} = \{\mathrm{a} \in \Phi(\mathrm{D}) |\ \mathrm{E}(b_1,\ldots,\mathrm{a},\ldots,b_n) \in \Phi(\mathcal{F})\}$$

where $\mathcal{F}$ is the appropriate category. For example, if $D \equiv N$:

$$\mathrm{I} = \{\mathrm{a} \in \Phi(\mathrm{N}) \mid \text{N-elim}(\mathrm{C},\mathrm{a},\mathrm{b},\mathrm{d}) \in \Phi(\text{app}(\mathrm{C},\mathrm{a}))\}$$

In the inner inductions, we want to show that $\Phi(D) \subseteq I$. By **lemma 6.3.1** on page 45 it is enough to show that I is closed under the generating rules. We notice that, for each **I**-rule, the corresponding Φ-rule ensures that we at least stay in Φ. To show that we stay in I, it is, according to **a.** in **lemma 6.3.2** on page 46 , sufficient to show that I **is closed under the introduction rules.** If in these cases

$$\mathrm{a} \equiv \text{intr}(c_1,\ldots,c_m)$$

and $c_1, \ldots, c_m$ are strongly normalizable, then, by **b.** in **lemma 6.3.2** on page 46 , it is enough to show that the resolution of the main redex of

$$\mathrm{E}(b_1,\ldots,\text{intr}(c_1,\ldots,c_m),\ldots,b_n)$$

is in $\Phi(\mathcal{F})$. This principle will be referred to as the **inner induction principle.** Thus, it says:

> In order to show that I is closed under the introduction rules, it suffices to show, under the assumptions of the rule, that c_1, ..., c_m are strongly normalizable and that if we reduce the main redex immediately (before any other reduction), then we end up in $\Phi(\mathcal{F})$.

7.2.2 Strong equality

All rules except **=7** and =8 trivially preserve ultra-correctness.

- **=7**
  The fact that this rule preserves ultra-correctness follows from **lemma 6.1.2** on page 38 .

- **=8**
  By the same reason as for **=7**.

7.2.3 General rules

- **judge-assumption:**
  This case is obvious.

- **judge-assumption-addition**
  This rule preserves ultra-correctness, since by **lemma A.1** on page 62 , JUDGE does not contain the variable x.

7.2.4 judge-cat

- **judge-cat-1**
  Obvious.

- **judge-cat-2**
  This case follows from **lemma 6.1.4** on page 38 .

- **judge-cat-2′**
  This case follows from the fact that **judge-cat-2** preserves ultra-correctness.

- **judge-cat-3**
  By **lemma 6.2.1** on page 41 applied to **OP-cat**.

- **judge-cat-3′**
  By assumption, *B cat (x ∈ A)* is ultra-correct. Hence, using that **judge-cat-3** preserves ultra-correctness, we get *(x ∈ A)B cat* is ultra-correct.

  Also, by assumption

  $$\text{A cat, A}' \text{ cat, B}' \text{ cat } (\text{x} \in \text{A})$$

  are ultra-correct. By **lemma 6.1.2** on page 38 , $\Phi(A) = \Phi(A')$ so that

  $$\text{a} \in \Phi(\text{A}') \implies \text{a} \in \Phi(\text{A}) \implies \text{B}'(\text{x:=a})\ \Phi\text{-cat}$$

  By **lemma A.12** on page 80 , *B′ cat (x ∈ A′)*. Thus, *B′ cat (x ∈ A′)* is ultra-correct, so that as above *(x ∈ A′)B′ cat* is ultra-correct.

- **judge-cat-4**
  By **lemma A.11** on page 79 , *B cat (x ∈ A)*. Since *b ∈ B (x ∈ A)* is ultra-correct, it follows that *A Φ-cat* and by **lemma 6.1.5** on page 39

  $$e \in \Phi(A) \implies B(x:=e)\ \Phi\text{-cat}$$

  Hence, by **OP-cat**, *(x ∈ A)B Φ-cat*. Now, we may use **abstr-Φ** to conclude that

  $$(x)b \in \Phi((x \in A)B)$$

  It follows that *(x)b ∈ (x ∈ A)B* is ultra-correct.

- **judge-cat-4′**
  This follows since **judge-cat-4** preserves ultra-correctness.

- **judge-cat-5 (elimination rule)**
  We have

  - $a \in \Phi(A)$
  - $b \in \Phi((x \in A)B)$
  - $e \in \Phi(A) \implies B(x:=e)\ \Phi\text{-cat}$

  We want to prove

  $$app(b,a) \in \Phi(B(x:=a))$$

  Since **judge-cat-3** preserves ultra-correctness, we have *(x ∈ A)B Φ-cat*. Hence, by **lemma 6.1.1** on page 37

  $$\Phi((x \in A)B) = \Phi((x \in \overline{A})\overline{B})$$

  IND(cat): In this case

  $$I = \{e \in \Phi((x \in \overline{A})\overline{B}) \mid app(e,a) \in \Phi(B(x:=a))\ \}$$

  The only introduction rule is
  abstrI:

  $$\frac{c \in \overline{B}\ (x \in \overline{A})\ ,\ d \in \Phi(\overline{A}) \implies c(x:=d) \in \Phi(\overline{B}(x:=d))}{(x)c \in I}$$

  By the **inner induction principle**, it suffices to show that c is strongly normalizable and that

  $$c(x:=a) \in \Phi(B(x:=a))$$

By **lemma 3.2.1** on page 18 and the usual argument with d as a constant, c is strongly normalizable. By assumption, $B(x{:=}a)$ Φ-*cat*. Hence, by **lemma 6.1.1** on page 37

$$\Phi(\mathrm{B}(\mathrm{x}{:=}\mathrm{a})) = \Phi(\overline{B}(\mathrm{x}{:=}\mathrm{a}))$$

Now the premisses in **abstrI** and the fact that $\Phi(A) = \Phi(\overline{A})$ give the result.

- **judge-cat-5′**
  Since **judge-cat-5** preserves ultra-correctness, we get that

  $$\mathrm{app}(\mathrm{b},\mathrm{a}) \in \mathrm{B}(\mathrm{x}{:=}\mathrm{a}) \quad \textit{and} \quad \mathrm{app}(\mathrm{b}',\mathrm{a}') \in \mathrm{B}(\mathrm{x}{:=}\mathrm{a}')$$

  are ultra-correct. Thus it is enough to prove that

  $$\Phi(\mathrm{B}(\mathrm{x}{:=}\mathrm{a})) = \Phi(\mathrm{B}(\mathrm{x}{:=}\mathrm{a}'))$$

  which is a consequence of the premisses, **lemma 6.1.2** on page 38 and **lemma A.13** on page 80 .

- **judge-cat-6-calc**
  Since **judge-cat-4** preserves ultra-correctness

  $$(\mathrm{x})\mathrm{b} \in (\mathrm{x} \in \mathrm{A})\mathrm{B} \textit{ is ultra-correct}$$

  By **lemma A.11** on page 79 , B *cat* $(x \in A)$. Since $b \in B$ $(x \in A)$ is ultra-correct, it follows as in **judge-cat-4** that B *cat* $(x \in A)$ is ultra-correct. Hence, since **judge-cat-5** preserves ultra-correctness, we get

  $$\mathrm{app}((\mathrm{x})\mathrm{b},\mathrm{a}) \in \mathrm{B}(\mathrm{x}{:=}\mathrm{a}) \textit{ is ultra-correct}$$

  Using $a \in \Phi(A)$ and that $b \in B$ $(x \in A)$ is ultra-correct, we get

  $$\mathrm{b}(\mathrm{x}{:=}\mathrm{a}) \in \mathrm{B}(\mathrm{x}{:=}\mathrm{a}) \textit{ is ultra-correct}$$

7.2.5 judge-N

- **judge-N-set**
  We have N Φ-*cat* and hence, by **lemma 6.1.4** on page 38 , $N \in \Phi(set)$.

- **judge-N-0**
  We want to prove that

  $$0 \in \Phi(\mathrm{N})$$

  This follows from **0**Φ, since N Φ-*cat*.

- **judge-N-s**
  This case follows as above from **s**Φ.

- **judge-N-elim**
  Suppose that the premisses in the rule are ultra-correct. We have to prove that

  $$\text{N-elim(C,a,b,d)} \in \Phi(\text{app(C,a)})$$

  IND(N): In this case

  $$\text{I} = \{\text{e} \in \Phi(\text{N}) \mid \text{N-elim(C,e,b,d)} \in \Phi(\text{app(C,e)})\}$$

  The introduction rules are
  0I:

  $$\frac{}{0 \in \text{I}}$$

  sI:

  $$\frac{\text{c} \in \text{I}}{\text{s(c)} \in \text{I}}$$

  the 0I case: By the **inner induction principle**, it suffices to show that

  $$\text{b} \in \Phi(\text{app(C,0)})$$

  which is true since $b \in app(C,0)$ is ultra-correct.

  the sI case: Now, c is in I, and hence it is strongly normalizable. Thus, by the **inner induction principle**, it suffices to show that

  $$\text{app(app(d,c),N-elim(C,c,b,d))} \in \Phi(\text{app(C,s(c))})$$

  Since $c \in I$

  $$\text{N-elim(C,c,b,d)} \in \Phi(\text{app(C,c)})$$

  Now, $c \in \Phi(N)$ so that, by **lemma 4.1** on page 20 , $c \in N$. Thus

  - $\text{c} \in \text{N}$
  - $\text{N-elim(C,c,b,d)} \in \text{app(C,c)}$

  are ultra-correct. There is a derivation of

  1. $\text{app(app(d,c),e)} \in \text{app(C,s(c))}$

  using only previous rules, the premisses and

  2. $\text{c} \in \text{N}$

  3. $\text{e} \in \text{app(C,c)}$

  Thus, if **2** and **3** are ultra-correct then so is **1**. Thus, putting $e \equiv N\text{-}elim(C,c,b,d)$, we are done.

- **judge-N-0-calc**
  This follows from the fact that **judge-N-elim** and **judge-N-0** preserve ultra-correctness.

- **judge-N-s-calc**
  We can prove that the judgements

  - N-elim(C,s(a),b,d) ∈ app(C,s(a))
  - app(app(d,a),N-elim(C,a,b,d)) ∈ app(C,s(a))

  are correct from the premisses using only the previous rules. Thus, they are also ultra-correct.

7.2.6 judge-Π

- **judge-Π-set**
  Suppose that the the premisses are ultra-correct. To prove that the conclusion is ultra-correct, it is, by **lemma 6.1.4** on page 38 , sufficient to prove that $\Pi(A,B)$ Φ-*cat*. To prove this we use **OP-Π**. By **lemma 6.1.4** , it is enough to prove that $(x \in A)app(B,x)$ Φ-*cat*. We use **OP-cat** to prove this. A simple derivation gives

  app(B,x) cat (x ∈ A)

  Let $a \in \Phi(A)$. Then, by **lemma 4.1** on page 20 , $a \in A$ is ultra-correct. By assumption, $B \in (x \in A)set$ is ultra-correct. Also, *set cat* $(x \in A)$ is ultra-correct. Thus, since **judge-cat-5** preserves ultra-correctness

  app(B,a) ϵ Φ(set)

  Hence, by **lemma 6.1.4**

  a ϵ Φ(A) $\Longrightarrow$ app(B,a) Φ-cat

  Thus, the premisses of **OP-cat** are satisfied and we are done.

- **judge-Π-λ**
  By λΦ, **lemma 3.2.1** on page 18 ,**lemma 6.1.4** and the just established fact that **judge-Π-set** preserves ultra-correctness.

- **judge-Π-elim**
  Suppose that the premisses in the rule are ultra-correct. By **lemma 6.1.4** and the fact that **judge-Π-set** preserves ultra-correctness, we have $\Pi(A,B)$ Φ-*cat*. Hence, by **lemma 6.1.1** on page 37

  $$\Phi(\Pi(A,B)) = \Phi(\Pi(\overline{A},\overline{B}))$$

  IND(Π) In this case

$$I = \{e \in \Phi(\Pi(\overline{A},\overline{B})) \mid \Pi\text{-elim}(A,B,C,e,d) \in \Phi(\text{app}(C,e)) \}$$

The only introduction rule is:

λI:

$$\frac{\begin{array}{l}\overline{A} = A' \in \text{set}\ ,\ \overline{B} = B' \in (x \in \overline{A})\text{set}\ , \\ A'\ \textbf{sn}\ ,\ B'\ \textbf{sn}\ , \\ c \in \Phi((x \in \overline{A})\text{app}(\overline{B},x))\end{array}}{\lambda(A',B',c) \in I}$$

By the **inner induction principle**, it suffices to prove that

$$\text{app}(d,c) \in \Phi(\text{app}(C,\lambda(A',B',c)))$$

Now

$$(x \in A)\text{app}(B,x)\ \text{cat}$$

has a derivation from the premisses using only previous rules. Thus it is also ultra-correct. By **lemma 6.1.1** on page 37

$$c \in \Phi((x \in A)\text{app}(B,x))$$

On the other hand, by **lemma 4.1** on page 20

$$c \in (x \in A)\text{app}(B,x)$$

is correct and hence ultra-correct. We can prove

$$\text{app}(d,c) \in \text{app}(C,\lambda(A,B,c))$$

using only $c \in (x \in A)app(B,x)$, previous rules and the premisses. Thus it is also ultra-correct. By $\lambda\Phi$

$$\lambda(A',B',c) \in \Pi(A,B)$$

is ultra-correct Furthermore, since **judge-Π-λ** preserves ultra-correctness

$$\lambda(A,B,c) \in \Pi(A,B)$$

is ultra-correct. There is a derivation of

1. $\text{app}(C,e)$ cat

using only previous rules, the premisses and

2. $e \in \Pi(A,B)$

Thus, if **2** is ultra-correct then so is **1**. Hence

- $\text{app}(C,\lambda(A,B,c))$ Φ-cat

- app(C,λ(A′,B′,c)) Φ-cat

On the other hand, by **lemma A.16** on page 84

app(C,λ(A,B,c)) = app(C,λ(A′,B′,c))

Now, by **lemma 6.1.2** on page 38

Φ(app(C,λ(A,B,c))) = Φ(app(C,λ(A′,B′,c)))

and we are done.

- **judge-Π-λ-calc**
  We can prove that the judgements
  - Π-elim(A,B,C,λ(A,B,b),d) ∈ app(C,λ(A,B,b))
  - app(d,b) ∈ app(C,λ(A,B,b))

  are correct from the premisses using only the previous rules. Thus they are also ultra-correct.

7.2.7 judge-Eq

- **judge-Eq-set**
  A direct consequence of **OP-Eq** and **lemma 6.1.4** .
- **judge-Eq-id**
  Since **judge-Eq** preserves ultra-correctness, the result follows from **idΦ** and **lemma 6.1.4** .
- **judge-Eq-elim**
  Suppose that the the premisses are ultra-correct. By **lemma 6.1.4** and the fact that **judge-Eq-set** preserves ultra-correctness, we have *Eq(A,a,b) Φ-cat*. Hence, by **lemma 6.1.1** on page 37

  Φ(Eq(A,a,b)) = Φ(Eq($\overline{A}$,$\overline{a}$,$\overline{b}$))

  IND(Eq) In this case

  I = {g ϵ Φ(Eq($\overline{A}$,$\overline{a}$,$\overline{b}$)) | Eq-elim(A,a,b,C,g,d) ϵ
  ϵ Φ(app(app(app(C,a),b),g)) }

  If $\overline{a} \not\equiv \overline{b}$ then there is no introduction rule, which immediately takes care of this case. Otherwise, the only introduction rule is

  idI:

$$\frac{\overline{A} = A' \in \text{set} \ , \ \overline{a} = c \in \overline{A} \ , \quad A' \ \textbf{sn} \ , \ c \ \epsilon \ \Phi(\overline{A})}{\text{id}(A',c) \ \epsilon \ I}$$

By the **inner induction principle**, it suffices to prove that

app(d,c) ϵ Φ(app(app(app(C,a),a),id(A′,c)))

Using **lemma 6.1.1** and the premisses we get in the usual way that

c $\in$ A

is ultra-correct. We can prove

app(d,c) $\in$ app(app(app(C,c),c),id(A,c))

using only $c \in A$, previous rules and the premisses. Thus it is also ultra-correct. Now

Eq(A,a,a) cat

is derivable from the premisses and previous rules and hence it is ultra-correct. Thus, by **idΦ**

id(A′,c) $\in$ Eq(A,a,a)

is ultra-correct. Furthermore, since **judge-Eq-id** preserves ultra-correctness

id(A,c) $\in$ Eq(A,c,c)

is ultra-correct. There is a derivation of

1. app(app(app(C,f),f),g) cat

using only previous rules, the premisses and

2. f $\in$ A
3. g $\in$ Eq(A,f,f)

Thus, if **2** and **3** are ultra-correct then so is **1**. Hence

- app(app(app(C,c),c),id(A,c)) Φ-cat
- app(app(app(C,a),a),id(A′,c)) Φ-cat

On the other hand, by **lemma A.16** on page 84

app(app(app(C,c),c),id(A,c)) = app(app(app(C,a),a),id(A′,c))

So that, by **lemma 6.1.2** on page 38

Φ(app(app(app(C,c),c),id(A,c))) = Φ(app(app(app(C,a),a),id(A′,c)))

and we are done.

- **judge-Eq-id-calc**
  By the same argument as in **judge-Π-λ-calc**.

7.2.8 judge-U

- **judge-U-set**
  Obvious.

- **judge-U-T**

  IND(U) In this case

  $$I = \{a \epsilon \Phi(U) \mid T(a) \epsilon \Phi(set)\}$$

  The only introduction rule is

  U-insertI:

  $$\frac{a \in U \text{ , } a \textbf{ intro} \text{ , } T(a)\ \Phi\text{-cat}}{a \epsilon I}$$

  That I is closed under this rule follows directly from **lemma 6.1.4** .

- **judge-U-n**
  We have $n \in U$ and n **intro**. By **U-insertΦ** and **lemma 6.1.4** , it suffices to prove that

  $$T(n) \epsilon \Phi(set)$$

  However, the only resolution of *T(n)* is

  $$T(n) \longrightarrow N$$

  and $N \epsilon \Phi(set)$. By **general-setΦ**, we are done.

- **judge-U-n-calc**
  We can prove that the judgements

  - $T(n) \in set$
  - $N \in set$

  are correct using only the previous rules. Thus they are also ultra-correct.

- **judge-U-π**
  By **U-insertΦ** and **lemma 6.1.4** , it suffices to prove that

  $$T(\pi(a,b)) \epsilon \Phi(set)$$

  The resolution of the main redex in $T(\pi(a,b))$ is

  $$T(\pi(a,b)) \longrightarrow \Pi(T(a),(x)T(app(b,x)))$$

  By **lemma 6.3.2** on page 46 , it suffices to prove that

$\Pi(T(a),(x)T(app(b,x))) \in \Phi(set)$

However

$\Pi(T(a),(x)T(app(b,x))) \in set$

has a derivation using the premisses and only previous rules. Thus it is ultra-correct and we are done.

- **judge-U-π-calc**
  By the usual argument.

- **judge-U-eq**
  By **U-insertΦ** and **lemma 6.1.4** , it suffices to prove

  $T(eq(a,b,c)) \in \Phi(set)$

  The resolution of the main redex in *T(eq(a,b,c))* is

  $T(eq(a,b,c)) \longrightarrow Eq(T(a),b,c)$

  By **lemma 6.3.2** , it is sufficient to prove that

  $Eq(T(a),b,c) \in \Phi(set)$

  However

  $Eq(T(a),b,c) \in set$

  has a derivation using the premisses and only previous rules. Thus it is ultra-correct and we are done.

- **judge-U-eq-calc**
  By the usual argument.

▓ (lemma 7.1)

A Appendix - Convertibility and equality

In this section, we will derive some properties of the formal system defined by the rules given in section 2 . We will use the following vector notations

$\mathbf{u} \in \mathbf{F} \equiv u_1 \in F_1, \ldots, u_p \in F_p \quad \mathrm{p} \geq 0$

$\mathbf{z} \in \mathbf{E} \equiv z_1 \in E_1, \ldots, z_m \in E_m \quad \mathrm{m} \geq 1$

$\mathbf{y} \in \mathbf{D} \equiv y_1 \in D_1, \ldots, y_r \in D_r \quad \mathrm{r} \geq 0$

$\mathbf{e} \in \mathbf{E} \equiv e_1 \in E_1 \, , e_2 \in E_2(z_1{:=}e_1), \ldots, e_m \in E_m(z_1{:=}e_1)\ldots(z_{m-1}{:=}e_{m-1})$

$\mathbf{e'} \in \mathbf{E} \equiv e_1{'} \in E_1 \, , e_2{'} \in E_2(z_1{:=}e_1{'}), \ldots, e_m{'} \in E_m(z_1{:=}e_1{'})\ldots(z_{m-1}{:=}e_{m-1}{'})$

$\mathbf{e{=}e'} \in \mathbf{E} \equiv e_1{=}e_1{'} \in E_1 \, , e_2{=}e_2{'} \in E_2(z_1{:=}e_1), \ldots, e_m{=}e_m{'} \in E_m(z_1{:=}e_1)\ldots(z_{m-1}{:=}e_{m-1})$

The names u_i, z_j ...denote different variables.

Definition A.1 (Correct assumption lists)

$(x_1 \in A_1, \ldots, x_n \in A_n)$

is said to be a correct assumption list if

A_1 *cat*
A_2 *cat* $(x_1 \in A_1)$
...
A_n *cat* $(x_1 \in A_1, \ldots, x_{n-1} \in A_{n-1})$

Note that there are no hidden assumptions in these judgements.

■ (definition A.1)

Lemma A.1 (Assumptions in judgements) *Suppose that*

JUDGE $(x_1 \in A_1, \ldots, x_n \in A_n)$

with no hidden assumptions. Then

$(x_1 \in A_1, \ldots, x_n \in A_n)$

is a correct assumption list and, if x *occurs in JUDGE, then* x *is one of the* x_i*:s.*

PROOF The following property is established by a straightforward induction. If

JUDGE $(x_1 \in A_1, \ldots, x_n \in A_n)$

then

A_1 cat
A_2 cat $(x_1 \in A_1)$
...
A_n cat $(x_1 \in A_1, \ldots, x_{n-1} \in A_{n-1})$

and, if x occurs in JUDGE, then x is one of the x_i:s and, if x occurs in A_j, then x is x_i for some i , $1 \leq i < j$.

∎ (lemma A.1)

Lemma A.2 (Inserted assumption) *Suppose that*

- *E cat*
- *JUDGE* $(\mathbf{y} \in \mathbf{D})$

then

JUDGE $(z \in E, \mathbf{y} \in \mathbf{D})$

PROOF By induction on the proof of JUDGE $(\mathbf{y} \in \mathbf{D})$. If the assumption list $\mathbf{y} \in \mathbf{D}$ is empty then the result follows from **judge-assumption-addition**. Otherwise, for each rule, the result follows directly from the induction hypothesis.

∎ (lemma A.2)

The following two lemmas are not needed in the rest of the paper, but have some interest of their own.

Lemma A.3 (Moving of an assumption) *If*

JUDGE $(\mathbf{u} \in \mathbf{F}, z \in E, w \in G, \mathbf{y} \in \mathbf{D})$

and

G cat

then

JUDGE $(\mathbf{u} \in \mathbf{F}, w \in G, z \in E, \mathbf{y} \in \mathbf{D})$

PROOF By induction on the proof of the judgement. If the rule does not expand the assumption list or if $\mathbf{y} \in \mathbf{D}$ is non-empty then the result follows directly by induction. Thus we only have to consider **judge-assumption** and **judge-assumption-addition** when $\mathbf{D}$ is empty.

judge-assumption

We have

$$\frac{\text{A cat } (\mathbf{u} \in \mathbf{F},\ z \in E)}{x \in A\ (\mathbf{u} \in \mathbf{F},\ z \in E,\ x \in A)}$$

where *A cat*. We have to prove

$$x \in A\ (\mathbf{u} \in \mathbf{F},\ x \in A,\ z \in E)$$

By **lemma A.1** and repeated use of **judge-assumption-addition**

$$A \text{ cat } (\mathbf{u} \in \mathbf{F})$$

and hence by **judge-assumption**, we get

$$x \in A\ (\mathbf{u} \in \mathbf{F},\ x \in A)$$

Furthermore, by **lemma A.1**

$$E \text{ cat } (\mathbf{u} \in \mathbf{F})$$

and hence, by **judge-assumption-addition**

$$E \text{ cat } (\mathbf{u} \in \mathbf{F},\ x \in A)$$

Thus, using **judge-assumption-addition** again, we obtain the result.

judge-assumption-addition

We have

$$\frac{\text{A cat } (\mathbf{u} \in \mathbf{F},\ z \in E)\ ,\ \text{JUDGE } (\mathbf{u} \in \mathbf{F},\ z \in E)}{\text{JUDGE } (\mathbf{u} \in \mathbf{F},\ z \in E,\ x \in A)}$$

where *A cat*. We have to prove

$$\text{JUDGE } (\mathbf{u} \in \mathbf{F},\ x \in A,\ z \in E)$$

As above

$$A \text{ cat } (\mathbf{u} \in \mathbf{F})$$

The result now follows from **lemma A.2** .

■ (lemma A.3)

Lemma A.4 (Permutation of assumptions) *If*

$$JUDGE\ (x_1 \in A_1,\ \ldots,\ x_n \in A_n)$$

and $(y_1 \in B_1, \ldots, y_n \in B_n)$ *is a permutation of* $(x_1 \in A_1, \ldots, x_n \in A_n)$ *such that* $(y_1 \in B_1, \ldots, y_n \in B_n)$ *still is a correct assumption list then*

$JUDGE\ (y_1 \in B_1, \ldots, y_n \in B_n)$

PROOF Since

B_1 cat

we may repeatedly use **lemma A.3** to move B_1 in the "A-list" to the first position. In this new judgement

B_2 cat $(y_1 \in B_1)$

and hence we may move it to the second position. By repeating this procedure, we get the result.

▓ (lemma A.4)

Lemma A.5 (Weak equals for equals in assumptions) *Suppose that*

- $JUDGE\ (x_1 \in A_1, \ldots, x_n \in A_n, \mathbf{y} \in \mathbf{D})$
- $A_1\ cat\ ,\ A_1'\ cat\ ,\ A_1 = A_1'$
- $A_2\ cat\ (x_1 \in A_1)\ ,\ A_2'\ cat\ (x_1 \in A_1')\ ,\ A_2 = A_2'\ (x_1 \in A_1)$
  ...
- $A_n\ cat\ (x_1 \in A_1, \ldots, x_{n-1} \in A_{n-1})$
  $A_n'\ cat\ (x_1 \in A_1', \ldots, x_{n-1} \in A_{n-1}')$
  $A_n = A_n'\ (x_1 \in A_1, \ldots, x_{n-1} \in A_{n-1})$

Then $JUDGE\ (x_1 \in A_1', \ldots, x_n \in A_n', \mathbf{y} \in \mathbf{D})$

PROOF By induction on n.

n = 1

We want to prove that

- JUDGE (z ∈ E, **y** ∈ **D**)
- E cat, E′ cat, E = E′

implies

- JUDGE (z ∈ E′, **y** ∈ **D**)

We prove this by induction on the derivation of JUDGE (z ∈ E, y ∈ D). We only need to check **judge-assumption** and **judge-assumption-addition**, since they are the only rules that extend the assumption list.

judge-assumption
When x is not z then the result follows directly from the induction hypothesis. Now, suppose that x is z. We have

$$\frac{\text{E cat}}{\text{x} \in \text{E (x} \in \text{E)}}$$

Since E' *cat*, we also get z ∈ E′ (z ∈ E′) and hence, by =**7**, z ∈ E (z ∈ E′).

judge-assumption-addition
Similarly, but easier.

induction step
Suppose $n > 1$ and the statement is true for n-*1*. We will now prove the statement for n. We have

JUDGE ($x_1 \in A_1'$, ..., $x_{n-1} \in A_{n-1}'$, $x_n \in A_n$, **y** ∈ **D**)

A_n cat ($x_1 \in A_1'$, ..., $x_{n-1} \in A_{n-1}'$)

A_n' cat ($x_1 \in A_1'$, ..., $x_{n-1} \in A_{n-1}'$)

$A_n = A_n'$ cat ($x_1 \in A_1'$, ..., $x_{n-1} \in A_{n-1}'$)

Hence, by the case $n = 1$

JUDGE ($x_1 \in A_1'$, ..., $x_{n-1} \in A_{n-1}'$, $x_n \in A_n'$, **y** ∈ **D**)

■ (lemma A.5)

Lemma A.6 (Substitution of expressions for variables in judgement)

Suppose that

- **e** ∈ **E** *(***u** ∈ **F***)*
- *JUDGE (***z** ∈ **E**, **y** ∈ **D***)*

Then

- *JUDGE(***z**:=**e***) (***u** ∈ **F**, **y** ∈ **D***(***z**:=**e***))*

As usual there may be a hidden assumption list to the left of the visible assumption list.

PROOF By induction on m, it is sufficient to prove that

- e ∈ E (**u** ∈ **F**)
- JUDGE (z ∈ E, **y** ∈ **D**)

implies

- JUDGE(z:=e) (**u** ∈ **F**, **y** ∈ **D**(z:=e))

This, in turn, is proved by induction on the derivation of *JUDGE (z ∈ E,* **y** *∈* **D***)*. A rule not changing the assumption list and not involving substitutions, is trivially verified. Thus we only have to check

> **judge-assumption , judge-assumption-addition , judge-cat-3 , judge-cat-3′ , judge-cat-4 , judge-cat-4′ , judge-cat-5 , judge-cat-5′** and **judge-cat-6-calc**

judge-assumption

We must consider two cases, x is z and x is not z.

x is not z

$$\frac{\text{A cat } (z \in E, \mathbf{y} \in \mathbf{D})}{x \in A\ (z \in E, \mathbf{y} \in \mathbf{D}, x \in A)}$$

By induction, we get *A(z:=e) cat (***u** ∈ **F**, **y** ∈ **D***(z:=e))*. We apply **judge-assumption** to get the result.

x is z

$$\frac{E \text{ cat}}{z \in E\ (z \in E)}$$

By assumption $e \in E$ *(***u** ∈ **F***)*, which is exactly what we need here.

judge-assumption-addition

Here we also have to consider the cases, x is z and x is not z.

x is not z

$$\frac{\text{A cat } (z \in E, \mathbf{y} \in \mathbf{D})\ ,\ \text{JUDGE } (z \in E, \mathbf{y} \in \mathbf{D})}{\text{JUDGE } (z \in E, \mathbf{y} \in \mathbf{D}, x \in A)}$$

By induction, we get

A(z:=e) cat (**u** ∈ **F**, **y** ∈ **D**(z:=e))
and
JUDGE(z:=e) (**u** ∈ **F**, **y** ∈ **D**(z:=e))

We apply **judge-assumption-addition** to get the result.

x is z

$$\frac{\text{E cat , JUDGE}}{\text{JUDGE (z} \in \text{E)}}$$

However, by **lemma A.1** on page 62 z is not in JUDGE and hence JUDGE is not changed when we replace z by e. Moreover, since e $\in$ E ($\mathbf{u} \in \mathbf{F}$), **lemma A.1** gives

F_1 cat
F_2 cat $(u_1 \in F_1)$
...
F_m cat $(u_1 \in F_1, \ldots, u_{m-1} \in F_{m-1})$

Thus, by successive use of **judge-assumption-addition**, we get JUDGE ($\mathbf{u} \in \mathbf{F}$).

judge-cat-3

$$\frac{\text{B cat (z} \in \text{E, } \mathbf{y} \in \mathbf{D}\text{, x} \in \text{A)}}{\text{(x} \in \text{A)B cat (z} \in \text{E, } \mathbf{y} \in \mathbf{D}\text{)}}$$

By induction, *B(z:=e) cat* ($\mathbf{u} \in \mathbf{F}$, $\mathbf{y} \in \mathbf{D}$*(z:=e)*, $x \in A$*(z:=e)*) is correct. Hence, by **judge-cat-3**, the result follows.

judge-cat-3′ , judge-cat-4 , judge-cat-4′
These rules are handled in the same way as **judge-cat-3**.

judge-cat-5

$$\frac{\begin{array}{l}\text{B cat (z} \in \text{E, } \mathbf{y} \in \mathbf{D}\text{, x} \in \text{A) ,} \\ \text{a} \in \text{A (z} \in \text{E, } \mathbf{y} \in \mathbf{D}\text{) , b} \in \text{(x} \in \text{A)B (z} \in \text{E, } \mathbf{y} \in \mathbf{D}\text{)}\end{array}}{\text{app(b,a)} \in \text{B(x:=a) (z} \in \text{E, } \mathbf{y} \in \mathbf{D}\text{)}}$$

By induction

- a(z:=e) $\in$ A(z:=e) ($\mathbf{u} \in \mathbf{F}$, $\mathbf{y} \in \mathbf{D}$(z:=e))
- b(z:=e) $\in$ (x $\in$ A(z:=e))B(z:=e) ($\mathbf{u} \in \mathbf{F}$, $\mathbf{y} \in \mathbf{D}$(z:=e))

Here x is chosen such that e does not contain x. Hence, by the same rule

app(b(z:=e),a(z:=e)) $\in$ (B(z:=e))(x:=a(z:=e)) ($\mathbf{u} \in \mathbf{F}$, $\mathbf{y} \in \mathbf{D}$(z:=e))

Since e does not contain x, we have

(B(z:=e))(x:=a(z:=e)) $\equiv$ (B(x:=a))(z:=e)

and the result follows.

judge-cat-5′

This rule is handled in the same way as **judge-cat-5**.

judge-cat-6-calc

$$\frac{a \in A\ (z \in E,\ \mathbf{y \in D})\ ,\ b \in B\ (z \in E,\ \mathbf{y \in D},\ x \in A)}{app((x)b,a) = b(x{:=}a) \in B(x{:=}a)\ (z \in E,\ \mathbf{y \in D})}$$

As above, the result follows from the commutativity of substitutions. However, in this case the fact that e does not contain x is a consequence of **lemma A.1** .

▓ (lemma A.6)

Definition A.2 (Weak congruence rule) *A judgement of one the following forms*

1. *B cat* $(\mathbf{z \in E},\ \mathbf{y \in D})$
2. $b \in B\ (\mathbf{z \in E},\ \mathbf{y \in D})$

is said to be **weakly congruent** *if*

- $\mathbf{e \in E}\ (\mathbf{u \in F})$
- $\mathbf{e' \in E}\ (\mathbf{u \in F})$
- $\mathbf{e{=}e' \in E}\ (\mathbf{u \in F})$

implies the following two properties

I.

$D_k(\mathbf{z{:=}e}) = D_k(\mathbf{z{:=}e'})\ (\mathbf{u \in F},\ y_1 \in D_1(\mathbf{z{:=}e}),\ \ldots,\ y_{k-1} \in D_{k-1}(\mathbf{z{:=}e}))$
$1 \leq k \leq r$

II.

1. $B(\mathbf{z{:=}e}) = B(\mathbf{z{:=}e'})\ (\mathbf{u \in F},\ \mathbf{y \in D}(\mathbf{z{:=}e}))$
2. $b(\mathbf{z{:=}e}) = b(\mathbf{z{:=}e'}) \in B(\mathbf{z{:=}e})\ (\mathbf{u \in F},\ \mathbf{y \in D}(\mathbf{z{:=}e}))$

in the respective cases.

A judgement of one of the other two (equality) forms are said to be weakly congruent without any extra requirements. A rule is said to **preserve weak congruence** *if for any instance of the rule, the weak congruence of the premisses implies the weak congruence of the conclusion.*

▓ (definition A.2)

Lemma A.7 (Weak congruence for basic rules) *Every rule, except possibly one of*

judge-N-s, judge-N-elim, judge-Π-set, judge-Π-λ, judge-Π-elim, judge-Eq-set, judge-Eq-id, judge-Eq-elim, judge-U-T, judge-U-π, judge-U-eq

preserves weak congruence.

PROOF

Equality rules
By definition.

Rules without premisses
Obvious, since they have no (hidden) assumptions.

=7

$$\frac{a \in A\ (z \in E, y \in D)\ ,\ A = B\ (z \in E, y \in D)}{a \in B\ (z \in E, y \in D)}$$

The condition on **D** follows from the left premiss and from the same premiss we also get

$$a(z:=e) = a(z:=e') \in A(z:=e)\ (u \in F, y \in D(z:=e))$$

By **lemma A.6** on page 66

$$A(z:=e) = B(z:=e)\ (u \in F, y \in D(z:=e))$$

Hence, **=8** gives the result.

judge-cat-2
This rule is taken care of in the same way as **=7**, by using **judge-cat-2′**.

judge-assumption
We must consider two cases, x is substituted and x is not substituted.

x is not substituted

$$\frac{A\ \text{cat}\ (z \in E, y \in D)}{x \in A\ (z \in E, y \in D, x \in A)}$$

I follows directly. **II** follows from **lemma A.6** , **judge-assumption** and **=1**.

x is substituted

$$\frac{\text{A cat } (\mathbf{z} \in \mathbf{E})}{\text{x} \in \text{A } (\mathbf{z} \in \mathbf{E}, \text{x} \in \text{A})}$$

I is vacuously satisfied. To verify **II** assume that

- $\mathbf{e} \in \mathbf{E}\ (\mathbf{u} \in \mathbf{F})$
- $\mathbf{e}' \in \mathbf{E}\ (\mathbf{u} \in \mathbf{F})$
- $\mathbf{e}{=}\mathbf{e}' \in \mathbf{E}\ (\mathbf{u} \in \mathbf{F})$
- $\text{a} \in \text{A}(\mathbf{z}{:=}\mathbf{e})\ (\mathbf{u} \in \mathbf{F})$
- $\text{a}' \in \text{A}(\mathbf{z}{:=}\mathbf{e}')\ (\mathbf{u} \in \mathbf{F})$
- $\text{a}{=}\text{a}' \in \text{A}(\mathbf{z}{:=}\mathbf{e})\ (\mathbf{u} \in \mathbf{F})$

But

$$\text{a}{=}\text{a}' \in \text{A}(\mathbf{z}{:=}\mathbf{e})\ (\mathbf{u} \in \mathbf{F})$$

is exactly what we have to prove, since, according to **lemma A.1** , A does not contain x.

judge-assumption-addition

We only treat the case when $JUDGE \equiv B\ cat$. The other cases are either similar or trivial.

x is not substituted

$$\frac{\text{A cat } (\mathbf{z} \in \mathbf{E}, \mathbf{y} \in \mathbf{D})\ ,\ \text{B cat } (\mathbf{z} \in \mathbf{E}, \mathbf{y} \in \mathbf{D})}{\text{B cat } (\mathbf{z} \in \mathbf{E}, \mathbf{y} \in \mathbf{D}, \text{x} \in \text{A})}$$

I. Follows directly.

II. Follows, using **lemma A.6** , **II** of the right hand premiss and **judge-assumption-addition**.

x is substituted

$$\frac{\text{A cat } (\mathbf{z} \in \mathbf{E})\ ,\ \text{B cat } (\mathbf{z} \in \mathbf{E})}{\text{B cat } (\mathbf{z} \in \mathbf{E}, \text{x} \in \text{A})}$$

I. Vacuously satisfied

II. Follows from **II** for the right premiss, since B does not contain x.

judge-cat-3

$$\frac{\text{B cat } (\mathbf{z} \in \mathbf{E}, \mathbf{y} \in \mathbf{D}, \text{x} \in \text{A})}{(\text{x} \in \text{A})\text{B cat } (\mathbf{z} \in \mathbf{E}, \mathbf{y} \in \mathbf{D})}$$

I. Follows directly.

II. By **I** for the premiss

$$A(\mathbf{z}:=\mathbf{e}) = A(\mathbf{z}:=\mathbf{e}')\ (\mathbf{u} \in \mathbf{F},\ \mathbf{y} \in \mathbf{D}(\mathbf{z}:=\mathbf{e}))$$

and by **II**

$$B(\mathbf{z}:=\mathbf{e}) = B(\mathbf{z}:=\mathbf{e}')\ (\mathbf{u} \in \mathbf{F},\ \mathbf{y} \in \mathbf{D}(\mathbf{z}:=\mathbf{e}),\ x \in A(\mathbf{z}:=\mathbf{e}))$$

Hence, by **judge-cat-3′** the result follows.

judge-cat-4

The same argument as for **judge-cat-3**.

judge-cat-5

$$\frac{\begin{array}{l} B \text{ cat } (\mathbf{z} \in \mathbf{E},\ \mathbf{y} \in \mathbf{D},\ x \in A)\ , \\ a \in A\ (\mathbf{z} \in \mathbf{E},\ \mathbf{y} \in \mathbf{D})\ ,\ b \in (x \in A)B\ (\mathbf{z} \in \mathbf{E},\ \mathbf{y} \in \mathbf{D}) \end{array}}{\text{app}(b,a) \in B(x:=a)\ (\mathbf{z} \in \mathbf{E},\ \mathbf{y} \in \mathbf{D})}$$

I. Follows directly.

II. By **II** for the premisses

$$a(\mathbf{z}:=\mathbf{e}) = a(\mathbf{z}:=\mathbf{e}') \in A(\mathbf{z}:=\mathbf{e})\ (\mathbf{u} \in \mathbf{F},\ \mathbf{y} \in \mathbf{D}(\mathbf{z}:=\mathbf{e}))$$
$$b(\mathbf{z}:=\mathbf{e}) = b(\mathbf{z}:=\mathbf{e}') \in (x \in A(\mathbf{z}:=\mathbf{e}))B(\mathbf{z}:=\mathbf{e})\ (\mathbf{u} \in \mathbf{F}, \mathbf{y} \in \mathbf{D}(\mathbf{z}:=\mathbf{e}))$$

Here x is chosen such that **e** does not contain x. By **judge-cat-5′**

$$\text{app}(b(\mathbf{z}:=\mathbf{e}),a(\mathbf{z}:=\mathbf{e})) = \text{app}(b(\mathbf{z}:=\mathbf{e}'),a(\mathbf{z}:=\mathbf{e}')) \in B(\mathbf{z}:=\mathbf{e})(x:=a(\mathbf{z}:=\mathbf{e}))$$
$$(\mathbf{u} \in \mathbf{F},\ \mathbf{y} \in \mathbf{D}(\mathbf{z}:=\mathbf{e}))$$

But, since **e** does not contain x

$$B(\mathbf{z}:=\mathbf{e})(x:=a(\mathbf{z}:=\mathbf{e})) \equiv B(x:=a)(\mathbf{z}:=\mathbf{e})$$

and the result follows.

■ (lemma A.7)

Lemma A.8 (Rules to judgements) *Consider the rules listed in* **lemma A.7** *. They are all of the following form*

$$\frac{a_1 \in A_1,\ \ldots,\ a_n \in A_n}{f(a_1,\ \ldots,\ a_n) \in A_{n+1}}$$

where $a_1, \ldots, a_n$ are meta-variables and $A_1, \ldots, A_{n+1}$ are expressions in the meta-variables. By abuse of names for formal variables, the following judgements are derivable using only rules which are listed in section 2 strictly before the rule at hand:

A_1 *cat*
A_2 *cat* $(a_1 \in A_1)$
...
A_{n+1} *cat* $(a_1 \in A_1, \ldots, a_n \in A_n)$

Furthermore, the following judgement is correct

$f(a_1, \ldots, a_n) \in A_{n+1} \quad (a_1 \in A_1, \ldots, a_n \in A_n)$

PROOF The first statement is proved by a straight-forward construction of derivations using only previous rules. We will give the details for **judge-Π-λ**.

$$\frac{\text{A} \in \text{set} \ , \ \text{B} \in (\text{x} \in \text{A})\text{set} \ , \ \text{b} \in (\text{x} \in \text{A})\text{app(B,x)}}{\lambda(\text{A,B,b}) \in \Pi(\text{A,B})}$$

By **judge-cat-1**

- set cat

By **judge-assumption-addition**, *set cat* $(A \in set)$. By **judge-assumption**, $A \in set$ $(A \in set)$ and by **judge-cat-2**

A cat (A ∈ set)

Hence, by **judge-assumption-addition**, *set cat* $(A \in set, x \in A)$. By **judge-cat-3**

- (x ∈ A)set cat (A ∈ set)

By **judge-assumption-addition**

A ∈ set (A ∈ set, B ∈ (x ∈ A)set)
A cat (A ∈ set, B ∈ (x ∈ A)set)
set cat (A ∈ set, B ∈ (x ∈ A)set)

and by **judge-assumption**

y ∈ A (A ∈ set, B ∈ (x ∈ A)set, y ∈ A)
B ∈ (x ∈ A)set (A ∈ set, B ∈ (x ∈ A)set)

By **judge-assumption-addition**

B ∈ (x ∈ A)set (A ∈ set, B ∈ (x ∈ A)set, y ∈ A)
set cat (A ∈ set, B ∈ (x ∈ A)set, y ∈ A, x ∈ A)

Hence, by **judge-cat-5**

app(B,y) ∈ set (A ∈ set, B ∈ (x ∈ A)set, y ∈ A)

and thus by **judge-cat-2** and **judge-cat-3**

- (y ∈ A)app(B,y) cat (A ∈ set, B ∈ (x ∈ A)set)

Hence by **judge-assumption-addition**

A ∈ set (A ∈ set, B ∈ (x ∈ A)set, b ∈ (y ∈ A)app(B,y))
B ∈ (x ∈ A)set (A ∈ set, B ∈ (x ∈ A)set, b ∈ (y ∈ A)app(B,y))

Finally, by **judge-Π-set** and **judge-cat-2**

- Π(A,B) cat (A ∈ set, B ∈ (x ∈ A)set, b ∈ (y ∈ A)app(B,y)

The second statement in the lemma follows from the first in the following way.

First apply **judge-assumption** to get $a_1 \in A_1$ *($a_1 \in A_1$)*. Then successively use **judge-assumption-addition** to get

$$a_1 \in A_1 \quad (a_1 \in A_1, \ldots, a_n \in A_n)$$

Again by **judge-assumption** $a_2 \in A_2$ *($a_1 \in A_1$, $a_2 \in A_2$)* and by **judge-assumption-addition**

$$a_2 \in A_2 \quad (a_1 \in A_1, \ldots, a_n \in A_n)$$

The same argument gives

$$a_i \in A_i \quad (a_1 \in A_1, \ldots, a_n \in A_n) \quad \text{for i} = 1, \ldots, \text{n}$$

Hence by **"judge-f"**

$$\text{f}(a_1, \ldots, a_n) \in A_{n+1} \quad (a_1 \in A_1, \ldots, a_n \in A_n)$$

■ (lemma A.8)

Lemma A.9 (Weak congruence) *All judgements are weakly congruent.*

PROOF We will prove that all rules preserve weak congruence, using **lemma A.7** on page 70 . The statement will then follow from an induction on the length of the derivation of the judgement. Suppose that all rules strictly before (according to the listing in section 2) the "f-rule" preserve weak congruence. We will prove that the same is true for the "f-rule"

$$\frac{a_1 \in A_1, \ldots, a_n \in A_n}{\text{f}(a_1, \ldots, a_n) \in A_{n+1}}$$

We will use the notation

$$\mathcal{A}_1, \ldots, \mathcal{A}_{n+1}$$

for the expressions obtained when we replace the schematic variables $a_1, \ldots, a_{n+1}$ by $x_1, \ldots, x_{n+1}$ in $A_1, \ldots, A_{n+1}$. By **lemma A.8** on page 72 , we have

$$\mathrm{f}(x_1, \ldots, x_n) \in \mathcal{A}_{n+1}\ (x_1 \in \mathcal{A}_1, \ldots, x_n \in \mathcal{A}_n)$$

(with no hidden assumptions). Note that

$$A_i \equiv \mathcal{A}_i(x_1{:=}a_1)\ldots(x_{i-1}{:=}a_{i-1}) \ \textit{for}\ \mathrm{i} = 1,\ldots, \mathrm{n{+}1}$$

We also introduce the notation

$$A_i' \equiv \mathcal{A}_i(x_1{:=}a_1')\ldots(x_{i-1}{:=}a_{i-1}') \ \textit{for}\ \mathrm{i} = 1,\ldots, \mathrm{n{+}1}$$

The first step in the proof is to prove that the "f-rule" has a companion "weak equality rule":

Suppose that for $i = 1, \ldots, n$

- $a_i \in A_i$
- $a_i' \in A_i'$
- $a_i{=}a_i' \in A_i$

then

- $\mathrm{f}(a_1, \ldots, a_n) = \mathrm{f}(a_1', \ldots, a_n') \in A_{n+1}$

To prove this, we prove the following by induction on k.

$$\begin{array}{l}(x_k)\ldots(x_n)\mathrm{f}(a_1, \ldots, a_{k-1}, x_k, \ldots, x_n) = \\ = (x_k)\ldots(x_n)\mathrm{f}(a_1', \ldots, a_{k-1}', x_k, \ldots, x_n) \in \\ \in ((x_k \in \mathcal{A}_k)\ldots(x_n \in \mathcal{A}_n)\mathcal{A}_{n+1})(x_1{:=}a_1)\ldots(x_{k-1}{:=}a_{k-1})\end{array}$$

For $k = 1$, it follows from **judge-cat-4** and **judge-cat-1**. Suppose the judgement is correct for k, we want to prove the corresponding judgement with $k{+}1$ instead of k. Put

$\mathbf{x} = x_1, \ldots, x_{k-1}$

$\mathbf{a} = a_1, \ldots, a_{k-1}$

$\mathbf{a}' = a_1', \ldots, a_{k-1}'$

$\mathrm{b} = (x_{k+1})\ldots(x_n)\mathrm{f}(\mathbf{a}, x_k, \ldots, x_n)$

$\mathrm{b}' = (x_{k+1})\ldots(x_n)\mathrm{f}(\mathbf{a}', x_k, \ldots, x_n)$

$\mathcal{B} = (x_{k+1} \in \mathcal{A}_k)\ldots(x_n \in \mathcal{A}_n)\mathcal{A}_{n+1}$

$\mathrm{B} = \mathcal{B}(\mathbf{x}{:=}\mathbf{a})$

$\mathrm{B}' = \mathcal{B}(\mathbf{x}{:=}\mathbf{a}')$

We have

$$a_k \in A_k,\ a_k' \in A_k',\ a_k = a_k' \in A_k \tag{1}$$

$$(x_k)b = (x_k)b' \in (x_k \in A_k)B \tag{2}$$

By **lemma A.8** on page 72

$\mathcal{B}$ cat $(x_1 \in \mathcal{A}_1, \ldots, x_k \in \mathcal{A}_k)$

Thus, by **lemma A.6** on page 66

B cat $(x_k \in A_k)$

Hence, by **judge-cat-5'**,

$$app((x_k)b, a_k) = app((x_k)b', a_k') \in B(x_k := a_k) \tag{3}$$

By repeated use of **judge-cat-4**

- $(x_{k+1})\ldots(x_n)\mathrm{f}(x_1, \ldots, x_n) \in \mathcal{B}$ $(x_1 \in \mathcal{A}_1, \ldots, x_k \in \mathcal{A}_k)$

Hence, by **lemma A.6** on page 66

$$b \in B\ (x_k \in A_k) \tag{4}$$

$$b' \in B'\ (x_k \in A_k') \tag{5}$$

Now, by **lemma A.8** on page 72 , the judgements (with no hidden assumptions)

- $\mathcal{B}$ cat $(x_1 \in \mathcal{A}_1, \ldots, x_k \in \mathcal{A}_k)$
- $\mathcal{A}_k$ cat $(x_1 \in \mathcal{A}_1, \ldots, x_{k-1} \in \mathcal{A}_{k-1})$

have derivations using rules up to the "f-rule". Hence, by assumption, the lemma holds, and we get

$$B = B'\ (x_k \in A_k) \tag{6}$$

$$A_k = A_k' \tag{7}$$

Observe that a_1, a_1', ..., may depend on hidden assumptions, $\mathbf{u} \in \mathbf{F}$, which in that case will occur as assumptions in 1,2,3,4,5,6,7.
Also, by **lemma A.6** , A_k *cat* and A_k' *cat*. Hence, by **lemma A.5** on page 65 and **=7**, we get

$\mathrm{b}' \in \mathrm{B}\ (x_k \in A_k)$

Hence

$\mathrm{b},\mathrm{b}' \in \mathrm{B}\ (x_k \in A_k)$

and by **judge-cat-6-calc**

- app((x_k)b,a_k) = b(x_k:=a_k) ∈ B(x_k:=a_k)
- app((x_k)b′,a_k′) = b′(x_k:=a_k′) ∈ B(x_k:=a_k)

Hence, by **=3** and **=5**

$$b(x_k:=a_k) = b'(x_k:=a_k') \in B(x_k:=a_k)$$

which is what we claimed. Hence, the induction step is completed, and we get

$$f(a_1, \ldots, a_n) = f(a_1', \ldots, a_n') \in A_{n+1}$$

The second step in the proof is to prove that the "f-rule" preserves weak congruence. Thus, assume

- **e ∈ E (u ∈ F)**
- **e′ ∈ E (u ∈ F)**
- **e=e′ ∈ E (u ∈ F)**

and assume

$$\frac{a_1 \in A_1(\mathbf{z} \in \mathbf{E}, \mathbf{y} \in \mathbf{D}), \ldots, a_n \in A_n(\mathbf{z} \in \mathbf{E}, \mathbf{y} \in \mathbf{D})}{f(a_1, \ldots, a_n) \in A_{n+1}(\mathbf{z} \in \mathbf{E}, \mathbf{y} \in \mathbf{D})}$$

is a correct instance of the "f-rule", i.e.

$$A_i \equiv \mathcal{A}_i(x_1:=a_1)\ldots(x_{i-1}:=a_{i-1}) \quad \textit{for i} = 1,\ldots,n$$

Assume that the premisses satisfy the "congruence conditions", i.e. for $i = 1,\ldots,n+1$

1. a_i(z:=e) = a_i(z:=e′) ∈ A_i(z:=e) **(u ∈ F, y ∈ D(z:=e))**

and for $k = 1,\ldots,r$

2. D_k(z:=e) = D_k(z:=e′) **(u ∈ F,** $y_1 \in D_1$(z:=e), …, $y_{k-1} \in D_{k-1}$(z:=e))

By **lemma A.6** , we have, for $i = 1,\ldots,n$

3. a_i(z:=e) ∈ A_i(z:=e) **(u ∈ F, y ∈ D(z:=e))**

4. a_i(z:=e′) ∈ A_i(z:=e′) **(u ∈ F, y ∈ D(z:=e′))**

We want to prove that **e′** in the assumption list of **4** above may be replaced by e. Since the premisses are correct, it follows from **lemma A.1** on page 62 that for $k = 1,\ldots,r$

D_k cat ($\mathbf{z} \in \mathbf{E}$, $y_1 \in D_1$, ..., $y_{k-1} \in D_{k-1}$)

By **lemma A.6** , for $k = 1,\ldots,r$

5. $D_k(\mathbf{z}:=\mathbf{e})$ cat ($\mathbf{u} \in \mathbf{F}$, $y_1 \in D_1(\mathbf{z}:=\mathbf{e})$, ..., $y_{k-1} \in D_{k-1}(\mathbf{z}:=\mathbf{e})$)

6. $D_k(\mathbf{z}:=\mathbf{e}')$ cat ($\mathbf{u} \in \mathbf{F}$, $y_1 \in D_1(\mathbf{z}:=\mathbf{e}')$, ..., $y_{k-1} \in D_{k-1}(\mathbf{z}:=\mathbf{e}')$)

Thus, by **2**, **5**, **6** and **lemma A.5** on page 65 , for $i = 1,\ldots,n$

7. $a_i(\mathbf{z}:=\mathbf{e}') \in A_i(\mathbf{z}:=\mathbf{e}')$ ($\mathbf{u} \in \mathbf{F}$, $\mathbf{y} \in \mathbf{D}(\mathbf{z}:=\mathbf{e})$)

Finally, **1**, **3**, **7** and the "equality f-rule" proved above give

- $\mathrm{f}(a_1(\mathbf{z}:=\mathbf{e}), \ldots, a_n(\mathbf{z}:=\mathbf{e})) = \mathrm{f}(a_1(\mathbf{z}:=\mathbf{e}'), \ldots, a_n(\mathbf{z}:=\mathbf{e}')) \in$
  $\in A_{n+1}(\mathbf{z}:=\mathbf{e})$ ($\mathbf{u} \in \mathbf{F}$, $\mathbf{y} \in \mathbf{D}(\mathbf{z}:=\mathbf{e})$)

∎ (lemma A.9)

Lemma A.10 (Reversed rule for categories) *If*

$(x \in A)B$ *cat*

then

B *cat* $(x \in A)$

PROOF We prove the statement by induction on the derivation. It is sufficient to take care of **judge-assumption-addition** and to prove that **judge-cat-2** is not applicable.

judge-assumption-addition

$$\frac{\mathrm{E\ cat}\ ,\ (\mathrm{x} \in \mathrm{A})\mathrm{B\ cat}}{(\mathrm{x} \in \mathrm{A})\mathrm{B\ cat}\ (\mathrm{z} \in \mathrm{E})}$$

By induction

B cat ($\mathrm{x} \in \mathrm{A}$)

Thus, by **lemma A.2** on page 63

B cat ($\mathrm{z} \in \mathrm{E}$, $\mathrm{x} \in \mathrm{A}$)

which is what we had to prove.

judge-cat-2

By an easy induction $(x \in A)B \in C$ is never correct, and hence $(x \in A)B \in set$ is never correct.

■ (lemma A.10)

Lemma A.11 (Equality and membership) *The following is true*

- $a \in A \implies A\ cat$
- $a = b \in A \implies A\ cat,\ a \in A,\ b \in A$
- $A = B \implies A\ cat,\ B\ cat$

PROOF By induction.

judge-cat-3′

$$\frac{A = A' \ , \ B = B' \ (x \in A)}{(x \in A)B = (x \in A')B'}$$

By induction, we get *B cat (x ∈ A)* and hence by **judge-cat-3**

(x ∈ A)B cat

Also by induction

B′ cat (x ∈ A)
A cat
A′ cat

Hence, by **lemma A.5** on page 65

B′ cat (x ∈ A′)

and then, by **judge-cat-3**

(x ∈ A′)B′ cat

judge-cat-5

$$\frac{B \text{ cat } (x \in A) \ , \ a \in A \ , \ b \in (x \in A)B}{app(b,a) \in B(x:=a)}$$

By **lemma A.6** on page 66 , *B(x:=a) cat*.

judge-cat-5′

$$\frac{B \text{ cat } x \in A \ , \ a = a' \in A \ , \ b = b' \in (x \in A)B}{app(b,a) = app(b',a') \in B(x:=a)}$$

By induction

a, a′ ∈ A, b,b′ ∈ (x ∈ A)B

As above *B(x:=a) cat*, and by **judge-cat-5**

app(b,a) ∈ B(x:=a), app(b′,a′) ∈ B(x:=a′)

Now, by **lemma A.9** , *B(x:=a) = B(x:=a′)* so that we get the result by using **=7**.

judge-cat-6-calc

$$\frac{\text{a} \in \text{A} \ , \ \text{b} \in \text{B} \ (\text{x} \in \text{A})}{\text{app((x)b,a)} = \text{b(x:=a)} \in \text{B(x:=a)}}$$

By induction, we have *B cat (x ∈ A)*. Hence, by **lemma A.6** *B(x:=a) cat*. By **judge-cat-4** and **judge-cat-5**

app((x)b,a) ∈ B(x:=a)

Finally, by **lemma A.6**

b(x:=a) ∈ B(x:=a)

The rest of the rules are easily checked.

∎ (lemma A.11)

Lemma A.12 (Equals for equals in assumptions) *Suppose*

- *JUDGE* $(x_1 \in A_1, \ldots, x_n \in A_n, \mathbf{y} \in \mathbf{D})$
- $A_1 = A_1'$
- $A_2 = A_2' \ (x_1 \in A_1)$

  ...
- $A_n = A_n' \ \ (x_1 \in A_1, \ldots, x_{n-1} \in A_{n-1})$

Then

$JUDGE \ (x_1 \in A_1', \ldots, x_n \in A_n', \mathbf{y} \in \mathbf{D})$

PROOF A direct consequence of **lemma A.11** and **lemma A.5** on page 65 .

∎ (lemma A.12)

Lemma A.13 (Congruence) *Substitution obeys the following laws*

- $\mathbf{e{=}e'} \in \mathbf{E}$ *(*$\mathbf{u} \in \mathbf{F}$*),* B *cat (*$\mathbf{z} \in \mathbf{E}$, $\mathbf{y} \in \mathbf{D}$*)* $\Longrightarrow$
  $B(\mathbf{z}{:=}\mathbf{e}) = B(\mathbf{z}{:=}\mathbf{e'})(\mathbf{u} \in \mathbf{F}, \mathbf{y} \in \mathbf{D}(\mathbf{z}{:=}\mathbf{e}))$
- $\mathbf{e{=}e'} \in \mathbf{E}$ *(*$\mathbf{u} \in \mathbf{F}$*),* $b \in B$ *(*$\mathbf{z} \in \mathbf{E}$, $\mathbf{y} \in \mathbf{D}$*)* $\Longrightarrow$
  $b(\mathbf{z}{:=}\mathbf{e}) = b(\mathbf{z}{:=}\mathbf{e'}) \in B(\mathbf{z}{:=}\mathbf{e})$ *(*$\mathbf{u} \in \mathbf{F}$, $\mathbf{y} \in \mathbf{D}(\mathbf{z}{:=}\mathbf{e})$*)*

PROOF A consequence of **lemma A.11** and **lemma A.9** on page 74 .

▓ (lemma A.13)

Lemma A.14 (Equality of categories) *If*

$C = (x \in A)B$

then C *may be represented as* $(x \in A')B'$*, where*

$A = A'$ *and* $B = B'$ $(x \in A)$

PROOF We show the following by induction on the correctness proof:

If $(x \in A)B = C$ or if $C = (x \in A)B$

then

C may be represented as $(x \in A')B'$,
where $A = A'$ and $B = B'$ $(x \in A)$

We only have to check the rules **=2, =4, =6, judge-assumption-addition, judge-cat-2′** and **judge-cat-3′**. The rules **=4** and **judge-cat-3′** are trivially checked.

=2

$$\frac{(x \in A)B \text{ cat}}{(x \in A)B = (x \in A)B}$$

By **lemma A.10** on page 78 , we have B *cat* $(x \in A)$ and by **lemma A.1** on page 62 A *cat*. Hence, by **=2**, $A = A$ and $B = B$ $(x \in A)$.

=6

We have two possibilities, handled in the same way. One of them is

$$\frac{(x \in A)B = D \ , \ D = C}{(x \in A)B = C}$$

By induction, D may be represented as

$(x \in A')B'$

where

$$A = A' \textit{ and } B = B' \ (x \in A)$$

Hence, also by induction, C may be represented as

$$(x \in A'')B''$$

where

$$A' = A'' \textit{ and } B' = B'' \ (x \in A')$$

By **lemma A.12**

$$B' = B'' \ (x \in A)$$

Thus, by **=6**

$$A = A'' \textit{ and } B = B'' \ (x \in A)$$

judge-assumption-addition
We have two possibilities, handled in the same way. One of them is

$$\frac{E \text{ cat }, C = (x \in A)B}{C = (x \in A)B \ (z \in E)}$$

By induction $C \equiv (x \in A')B'$, where

$$A = A'$$
$$B = B' \ (x \in A)$$

Thus, by **lemma A.2**

$$A = A' \ (z \in E)$$
$$B = B' \ (z \in E, x \in A)$$

judge-cat-2′
By an easy induction $(x \in A)B = C \in D$ is never correct, and hence $(x \in A)B = C \in set$ is never correct.

∎ (lemma A.14)

Lemma A.15 (Parameters of forms) *We have the following "reversed stability" of forms for correct judgements*

- $(x)b \in (x \in A)B \implies b \in B \ (x \in A)$
- $s(a) \in C \implies a \in N$

- $\lambda(A,B,b) \in C \implies b \in (x \in A)app(B,x)\ ,\ C = \Pi(A,B)$
- $\Pi(A,B) \in C \implies A \in set,\ B \in (x \in A)set$
- $\Pi(A,B)\ cat \implies A \in set,\ B \in (x \in A)set$
- $Eq(A,a,b) \in C \implies A \in set,\ a \in A,\ b \in A$
- $Eq(A,a,b)\ cat \implies A \in set,\ a \in A,\ b \in A$
- $id(A,a) \in C \implies a \in A\ ,\ C = Eq(A,a,a)$
- $\pi(a,b) \in C \implies a \in U\ ,\ b \in (x \in T(a))U$
- $eq(a,b,c) \in C \implies a \in U\ ,\ b \in T(a)\ ,\ c \in T(a)$

PROOF By induction on the proof of a judgement. We consider the form $(x)b$. All the other forms are trivially handled. (Note that, if the induction hypothesis for e.g. s had been

$$s(a) \in N \implies a \in N$$

then the rule **=7** would create problems.) The judgement $(x)b \in (x \in A)B$ is proved by one of the rules **=7** , **judge-assumption-addition** , **judge-cat-4**. Obviously, the conclusion follows if the last rule is used.

=7

$$\frac{(x)b \in C\ ,\ C = (x \in A)B}{(x)b \in (x \in A)B}$$

By **lemma A.14** , C may be represented as

$$(x \in A')B'$$

where

$$A = A' \text{ and } B = B'\ (x \in A)$$

By induction

$$b \in B'\ (x \in A')$$

and **lemma A.12** gives

$$b \in B'\ (x \in A)$$

and finally, by **=7**

$$b \in B\ (x \in A)$$

judge-assumption-addition

$$\frac{\text{E cat , (x)b} \in \text{(x} \in \text{A)B}}{\text{(x)b} \in \text{(x} \in \text{A)B (z} \in \text{E)}}$$

By induction

b ∈ B (x ∈ A)

Hence, by **lemma A.2**

b ∈ B (z ∈ E, x ∈ A)

∎ (lemma A.15)

Lemma A.16 (Equality rules) *Consider a rule listed in* **lemma A.7** *. It is of the following form*

$$\frac{a_1 \in A_1,\ \ldots,\ a_n \in A_n}{f(a_1,\ \ldots,\ a_n) \in A_{n+1}}$$

It satisfies

$$a_1 = a_1' \in A_1,\ \ldots,\ a_n = a_n' \in A_n \implies$$
$$\implies f(a_1,\ \ldots,\ a_n) = f(a_1',\ \ldots,\ a_n') \in A_{n+1}$$

PROOF According to **lemma A.11** on page 79 and the first part of the proof of **lemma A.9** on page 74 .

∎ (lemma A.16)

Lemma A.17 (Consistency) *We have the following "consistency" properties*

- $\Pi(A,B) = \Pi(A',B') \implies A = A' \in set\ ,\ B = B' \in (x \in A)set$
- $Eq(A,a,b) = Eq(A',a',b') \implies A = A' \in set\ ,\ a = a' \in A\ ,\ b = b' \in A$

PROOF With a technique similar to the one used in [Coq90].

∎ (lemma A.17)

Lemma A.18 (Reduction implies equality) *Suppose that* a ∈ A *or* a cat *is a correct judgement and that* $a \xrightarrow{*} b$. *Then* a = b ∈ A *or* a = b , *respectively.*

PROOF It is sufficient to prove the statement for $a \longrightarrow b$. We make the following induction hypothesis on correct judgements:

> For every JUDGE $(x_1 \in A_1, \ldots, x_n \in A_n)$ we require the assumptions to satisfy
>
> $A_i \longrightarrow A_i' \implies A_i = A_i'$ $(x_1 \in A_1, \ldots, x_{i-1} \in A_{i-1})$
> for $1 \leq i \leq n$
>
> Furthermore, if "*JUDGE*" is "$a \in A$" or "*a cat*" then we require $a \longrightarrow b$ to imply $a = b \in A$ $(x_1 \in A_1, \ldots, x_n \in A_n)$ or $a = b$ $(x_1 \in A_1, \ldots, x_n \in A_n)$, respectively. For the other two (equality) judgement forms there is no extra requirement.

The rules **judge-cat-2** and **judge-cat-4** follow by induction using **judge-cat-2'** and **judge-cat-4'**, respectively. Using **lemma A.16** on page 84 the same argument applies to the rest of the relevant "introduction"-rules. In the following we only consider the rules that not trivially preserve the requirements.

=7
: In this case, if $a \longrightarrow b$ then, by induction, $a = b \in A$ and hence by **=8** we get $a = b \in B$.

judge-assumption. Since x has no reductions, we only have to check that the extended assumption list satisfies the requirements. However, this follows directly by using the induction hypothesis on A.

judge-assumption-addition. Basically the same argument as in **judge-assumption** handles this case.

judge-cat-3
: The assumption list has shrunk, so there is no trouble with that. We have the following possibilities for the reduction

  1. (x ∈ A)B ⟶ (x ∈ A′)B , where A ⟶ A′
  2. (x ∈ A)B ⟶ (x ∈ A)B′ , where B ⟶ B′

  Case **1** follows by the induction hypothesis and **=2**. The same goes for **2**, noting that A *cat* by **lemma A.1** on page 62 so that **=2** gives $A = A$.

judge-cat-5
: We have the following possibilities for the reduction

  1. app(b,a) ⟶ app(b′,a) , where b ⟶ b′
  2. app(b,a) ⟶ app(b,a′) , where a ⟶ a′

3. b $\equiv$ (x)c and app((x)c,a) $\longrightarrow$ c(x:=a)

Case **1** and **2** follow by induction, using judge-cat-5′. In case **3**, we know that *(x)c* $\in$ *(x* $\in$ *A)B* and hence, by **lemma A.15** , *c* $\in$ *B (x* $\in$ *A)*. Thus, by **judge-cat-6-calc**, *app((x)c,a) = c(x:=a)* $\in$ *B(x:=a)*.

judge-N-elim

We have the following possibilities for the reduction

1. N-elim(C,a,b,d) $\longrightarrow$ N-elim(C′,a,b,d) , where C $\longrightarrow$ C′

2. N-elim(C,a,b,d) $\longrightarrow$ N-elim(C,a′,b,d) , where a $\longrightarrow$ a′

3. N-elim(C,a,b,d) $\longrightarrow$ N-elim(C,a,b′,d) , where b $\longrightarrow$ b′

4. N-elim(C,a,b,d) $\longrightarrow$ N-elim(C,a,b,d′) , where d $\longrightarrow$ d′

5. a $\equiv$ 0 *and* N-elim(C,0,b,d) $\longrightarrow$ b

6. a $\equiv$ s(e) *and* N-elim(C,s(e),b,d) $\longrightarrow$ app(app(d,e),N-elim(C,e,b,d))

The cases **1 - 4** follow by induction, using **lemma A.16** . In the case of **5**, the conclusion follows from **judge-N-0-calc**. In the case of **6**, we use **lemma A.15** to deduce that *e* $\in$ *N* and hence **judge-N-s-calc** gives the result.

judge-Π-elim

We have the following possibilities for the reduction

1. Π-elim(A,B,C,a,d) $\longrightarrow$ Π-elim(A′,B,C,a,d) , where A $\longrightarrow$ A′

2. Π-elim(A,B,C,a,d) $\longrightarrow$ Π-elim(A,B′,C,a,d) , where B $\longrightarrow$ B′

3. Π-elim(A,B,C,a,d) $\longrightarrow$ Π-elim(A,B,C′,a,d) , where C $\longrightarrow$ C′

4. Π-elim(A,B,C,a,d) $\longrightarrow$ Π-elim(A,B,C,a′,d) , where a $\longrightarrow$ a′

5. Π-elim(A,B,C,a,d) $\longrightarrow$ Π-elim(A,B,C,a,d′) , where d $\longrightarrow$ d′

6. a $\equiv$ λ(A′,B′,b) *and* Π-elim(A,B,C,a,d) $\longrightarrow$ app(d,b)

The cases **1 - 5** follow by induction, using **lemma A.16** . In the case of **6**, λ(A′,B′,b) $\in$ Π(A,B) and we use **lemma A.15** to deduce that

b $\in$ (x $\in$ A′)app(B′)
Π(A′,B′) = Π(A,B)

By **lemma A.17**

A = A′ $\in$ set , B = B′ $\in$ (x $\in$ A)set

and hence , by **lemma A.16**

λ(A,B,b) = λ(A′,B′,b) $\in$ Π(A,B)

so that

$$\Pi\text{-elim}(A,B,C,a,d) \equiv \Pi\text{-elim}(A,B,C,\lambda(A',B',b),d) =$$
$$= \Pi\text{-elim}(A,B,C,\lambda(A,B,b),d) \in \text{app}(C,a)$$

$$\Pi\text{-elim}(A,B,C,\lambda(A,B,b),d) = \text{app}(d,b) \in \text{app}(C,\lambda(A,B,b))$$

By **judge-cat-5′**

$$\text{app}(C,\lambda(A,B,b)) = \text{app}(C,\lambda(A',B',b)) \in \text{set}$$

hence by **judge-cat-2′**, **=8** and **=5**

$$\Pi\text{-elim}(A,B,C,a,d) = \text{app}(d,b) \in \text{app}(C,a)$$

judge-Eq-elim

We have the following possibilities for the reduction

1. $\text{Eq-elim}(A,a,b,C,g,d) \longrightarrow \text{Eq-elim}(A',a,b,C,g,d)$, where $A \longrightarrow A'$
2. $\text{Eq-elim}(A,a,b,C,g,d) \longrightarrow \text{Eq-elim}(A,a',b,C,g,d)$, where $a \longrightarrow a'$
3. $\text{Eq-elim}(A,a,b,C,g,d) \longrightarrow \text{Eq-elim}(A,a,b',C,g,d)$, where $b \longrightarrow b'$
4. $\text{Eq-elim}(A,a,b,C,g,d) \longrightarrow \text{Eq-elim}(A,a,b,C',g,d)$, where $C \longrightarrow C'$
5. $\text{Eq-elim}(A,a,b,C,g,d) \longrightarrow \text{Eq-elim}(A,a,b,C,g',d)$, where $g \longrightarrow g'$
6. $\text{Eq-elim}(A,a,b,C,g,d) \longrightarrow \text{Eq-elim}(A,a,b,C,g,d')$, where $d \longrightarrow d'$
7. $g \equiv \text{id}(A',e)$ *and* $\text{Eq-elim}(A,a,b,C,g,d) \longrightarrow \text{app}(d,e)$

The cases **1 - 6** follow by induction, using **lemma A.16** . In the case of **7**

$$\text{id}(A',e) \in \text{Eq}(A,a,b)$$

and hence, by **lemma A.15**

$$e \in A' \ , \ \text{Eq}(A',e,e) = \text{Eq}(A,a,b)$$

Thus, by **lemma A.17**

$$A' = A \in \text{set} \ , \ e = a \in A \ , \ e = b \in A$$

so that by **lemma A.16**

$$\text{Eq-elim}(A,a,b,C,g,d) \equiv \text{Eq-elim}(A,a,b,C,\text{id}(A',e),d) =$$
$$= \text{Eq-elim}(A,e,e,C,\text{id}(A,e) \in \text{app}(\text{app}(\text{app}(C,a),b),g)$$

$$\text{Eq-elim}(A,e,e,C,\text{id}(A,e) = \text{app}(d,e) \in \text{app}(\text{app}(\text{app}(C,e),e),\text{id}(A,e))$$

Using **judge-cat-5′** repeatedly, we get

$$\text{app}(\text{app}(\text{app}(C,a),b),g) = \text{app}(\text{app}(\text{app}(C,e),e),\text{id}(A,e)) \in \text{set}$$

hence by **judge-cat-2′**, **=8** and **=5**

$$\text{Eq-elim}(A,a,b,C,g,d) = \text{app}(d,b) \in \text{app}(\text{app}(\text{app}(C,a),b),g)$$

judge-T-U

We have the following possibilities for the reduction

1. $T(a) \longrightarrow T(a')$, where $a \longrightarrow a'$
2. $a \equiv n$ *and* $T(a) \longrightarrow N$
3. $a \equiv \pi(a,b)$ *and* $T(a) \longrightarrow \Pi(T(a),(x)T(\text{app}(b,x))$
4. $a \equiv \text{eq}(b,c,d)$ *and* $T(a) \longrightarrow \text{Eq}(T(b),c,d)$

Case **1** follows by induction, using **lemma A.16** . Case **2** is trivial. In the case of **3**

$$\pi(a,b) \in U$$

and hence, by **lemma A.15**

$$a \in U \text{ , } b \in (x \in T(a))U$$

so that

$$T(a) \equiv T(\pi(a,b)) = \Pi(T(a),(x)T(\text{app}(b,x)) \in \text{set}$$

In the case of **4**

$$\text{eq}(b,c,d) \in U$$

and hence, by **lemma A.15**

$$b \in U \text{ , } c \in T(a) \text{ , } d \in T(a)$$

so that

$$T(a) \equiv T(\text{eq}(b,c,d)) = \text{Eq}(T(b),c,d) \in \text{set}$$

The rest of the relevant rules are all on "introduction" form and follow by induction using **lemma A.16** on page 84 .

∎ (lemma A.18)

Lemma A.19 (Reduction and membership) *Suppose that*

$a \in A$ *(*or *a cat)*

and that

$a \longrightarrow b$

then

$b \in A$ *(*or b *cat)*

PROOF A direct consequence of **lemma A.18** and **lemma A.11** .

▓ (lemma A.19)

Lemma A.20 (Conversion and Equality) *We have the following*

- A *cat,* B *cat,* A *conv* $B \iff A = B$
- $a \in A$, $b \in A$, a *conv* $b \iff a = b \in A$

PROOF

$\Longrightarrow$:
A direct consequence of **lemma A.18** and **lemma 2.3.1** on page 12 .

$\Longleftarrow$:
By **lemma A.11** and an easy induction.

▓ (lemma A.20)

Acknowledgement

We are indebted to Per Martin-Löf and Dag Prawitz, who explained the principles for establishing a normalization proof.

References

[Bis67] Erret Bishop. *Foundations of Constructive Analysis.* McGraw-Hill, 1967.

[Coq90] Thierry Coquand. An algorithm for testing conversion in type theory. In G. Huet and G. Plotkin, editors, *Proceedings of the first workshop on logical frameworks*, pages 135–144, 1990.

[Gir71] Jean-Yves Girard. Une extension de l'intérpretation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types. In J.E. Fenstad, editor, *Proceedings of the second Scandinavian Logic Symposium*, pages 63–92. North-Holland, 1971.

[How80] W.A. Howard. The formulae-as-types notion of construction. In J.P. Seldin and J.R. Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490, London, 1980. Academic Press.

[ML71a] Per Martin-Löf. Hauptsatz for the intuitionistic theory of iterated inductive definitions. In J.E. Fenstad, editor, *Proceedings of the second Scandinavian Logic Symposium*, pages 179–216. North-Holland, 1971.

[ML71b] Per Martin-Löf. Hauptsatz for the theory of species. In J.E. Fenstad, editor, *Proceedings of the second Scandinavian Logic Symposium*, pages 217–233. North-Holland, 1971.

[ML72] Per Martin-Löf. An intuitionistic theory of types. Unpublished manuscript, University of Stockholm, 1972.

[Pra71] Dag Prawitz. Ideas and results in proof theory. In J.E. Fenstad, editor, *Proceedings of the second Scandinavian Logic Symposium*, pages 235–307. North-Holland, 1971.

[Sve90] Catarina Svensson. A normalization proof for Martin-Löf's type theory. Dissertation for the Licentiate Degree in Computer Science at University of Göteborg, 1990.

[Tai67] W.W. Tait. Intensional interpretation of functionals of finite type **I**. *The Journal of Symbolic Logic*, 32:198–212, 1967.